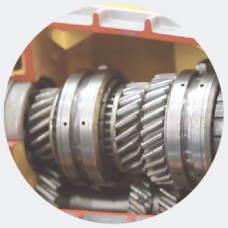
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA E GESTÃO INDUSTRIAL






Students association management - strategy formulation and support tools

JAKUB KARLO
Junho de 2021

POLITÉCNICO
DO PORTO

Students association management - strategy formulation and support tools

Master Thesis

Jakub Karło

24.06.2021

Supervisors:
Carlos Ferreira
Paulo Proenca

Contents

1 Introduction **1**
- 1.1 Problem statement . . . 1
- 1.2 Solution . . . 1
- 1.3 Goal and scope of the thesis . . . 1
- 1.4 Document structure . . . 2

2 State of the art **3**
- 2.1 Strategy development tools . . . 3
  - 2.1.1 Mission and Vision Statement . . . 3
  - 2.1.2 PEST Analysis . . . 4
  - 2.1.3 SWOT Analysis . . . 4
  - 2.1.4 SWOT matrix . . . 5
  - 2.1.5 Long-term objectives . . . 7
  - 2.1.6 Balanced scorecard . . . 8
- 2.2 Existing applications . . . 9
  - 2.2.1 Monday.com . . . 10
  - 2.2.2 Jedox . . . 13
  - 2.2.3 Wrike . . . 14
  - 2.2.4 ESM+Strategy . . . 15
  - 2.2.5 Conclusion . . . 18
- 2.3 Server-side and client-side applications . . . 18
- 2.4 Backend technologies . . . 19
  - 2.4.1 Enterprise Java Beans - EJB . . . 19
  - 2.4.2 Spring . . . 20
  - 2.4.3 Spring Boot . . . 21
  - 2.4.4 Framework choice . . . 21
  - 2.4.5 Java . . . 22
  - 2.4.6 Relational databases . . . 22
  - 2.4.7 Non-relational databases . . . 23
  - 2.4.8 Apache Tomcat . . . 24
- 2.5 Frontend technologies . . . 24
  - 2.5.1 AngularJS . . . 24
  - 2.5.2 React.js . . . 25
  - 2.5.3 Thymeleaf . . . 25
  - 2.5.4 Framework choice . . . 26
- 2.6 Summary . . . 26

3 Project description **27**
3.1 Requirements identification . . . 27
3.1.1 Functional . . . 27
3.1.2 Non-functional . . . 28
3.2 Solution vision . . . 28
3.2.1 API . . . 29
3.2.2 Frontend . . . 29
3.3 Domain model . . . 31
3.4 Design . . . 33
3.4.1 Class diagram . . . 33
3.4.2 Sequence diagram . . . 34

4 Implementation **37**
4.1 Strategy generation . . . 37
4.2 List management . . . 40

5 User Guide **44**
5.1 Login . . . 44
5.2 Strategy list . . . 45
5.3 Strategy dashboard . . . 47
5.4 Mission and Vision . . . 50
5.5 PEST analysis . . . 51
5.6 SWOT analysis . . . 53
5.7 SWOT matrix . . . 55
5.8 Long term objectives . . . 58
5.9 Objective domains . . . 60
5.10 Generate document . . . 61

6 Strategy development steps **62**
6.1 Mission and Vision . . . 62
6.1.1 ESN-EYE Mission . . . 62
6.1.2 ESN Vision . . . 62
6.1.3 ESN-EYE Vision . . . 62
6.2 Consultations with members . . . 62
6.3 PEST Analysis . . . 63
6.3.1 Political factors . . . 63
6.3.2 Economical factors . . . 64
6.3.3 Social factors . . . 64

6.3.4 Technological factors 65
6.4 SWOT Analysis 65
6.4.1 Strengths 66
6.4.2 Weaknesses 66
6.4.3 Opportunities 67
6.4.4 Threats 67
6.5 SWOT Matrix 68
6.6 Long-term Objectives 69
6.6.1 Organisation role among the local society 69
6.6.2 Partnerships and collaborations 70
6.6.3 Members and their development 70
6.6.4 Administration and finance 71
6.6.5 Promotion 71

7 Summary **72**

List of Figures

1 Exemplary SWOT matrix with ratings [Source: Own work] . . 6
2 Balanced scorecard for Royal Dutch Shell company [Source: Own work] 9
3 Template center screen [Source: Monday.com] 11
4 Competitors analysis template [Source: Monday.com] 11
5 Template adjusted for the needs of scorecard and KPI monitoring [Source: Monday.com] 12
6 Jedox application homepage [Source: Jedox] 13
7 Wrike application homepage [Source: Wrike] 14
8 ESM pricing page [Source: ESM] 15
9 ESM+Strategy application homepage [Source: ESM] 16
10 ESM+Strategy PESTEL analysis page [Source: ESM] 17
11 ESM+Strategy SWOT analysis page [Source: ESM] 17
12 Spring Service structure [Source: Pinterest [15]] 20
13 Downloads number of Thymeleaf from Maven Central repository between February 2020 and January 2021 [Source: Thymeleaf Official Webpage [30]] 26
14 Tradic color schema [Source: Canva.com [31]] 30
15 Application basic colours [Source: Webfx.com [33]] 31
16 Domain model [Source: Own work] 32
17 Class diagram of the application [Source: Own work] 33
18 PEST Analysis sequence diagram [Source: Own work] 35
19 *generateDocument()* method code fragment [Source: Own work] 38
20 Code fragment responsible for sorting the objectives [Source: Own work] 39
21 File generated using Apache POI [Source: Own work] 40
22 *PestFactorsWrapper* class code [Source: Own work] 41
23 *pest_factors_political.html* file code [Source: Own work] 41
24 *StrategyController* code [Source: Own work] 42
25 *preparePestFactors()* method code [Source: Own work] 43
26 Login page [Source: Own work] 44
27 Welcome page [Source: Own work] 45
28 Strategy list page [Source: Own work] 45
29 New strategy page [Source: Own work] 46
30 Strategy list page with new strategy added [Source: Own work] 47
31 Strategy dashboard page [Source: Own work] 48

32 Strategy removal confirmation modal [Source: Own work] . . . 49
33 Strategy dashboard tile buttons [Source: Own work] 49
34 Mission and vision page [Source: Own work] 50
35 Strategy dashboard with sidebar hints [Source: Own work] . . 51
36 PEST analysis dashboard [Source: Own work] 51
37 PEST factors page [Source: Own work] 52
38 PEST factors overview [Source: Own work] 53
39 SWOT analysis dashboard [Source: Own work] 54
40 SWOT factors page [Source: Own work] 54
41 SWOT factors overview [Source: Own work] 55
42 SWOT matrix creation confirmation dialog [Source: Own work] 56
43 SWOT matrix dashboard [Source: Own work] 56
44 SWOT matrix affection levels page [Source: Own work] 57
45 SWOT matrix solutions page [Source: Own work] 58
46 Long term objectives page [Source: Own work] 59
47 Objective domains page [Source: Own work] 60
48 Saving document prompt [Source: Own work] 61

1 Introduction

In this section there is a problem defined alongside with the solution. There is also presented the goal and scope of the thesis, as well as the document structure.

1.1 Problem statement

The Erasmus Students Network department in Lodz has created their first operation strategy. The process has lasted significantly long time and required a time-consuming research about strategic management principles. As a consequence, the team responsible for strategy creation has gained sufficient skills to develop it next time faster and more competently. However, the nature of ESN-EYE Lodz is that the members exchange occurs very often and the probability that the next team would have to go all over exactly the same learning process is pretty high. They need to find a way to save time and gather all necessary knowledge in one place.

1.2 Solution

Considering the previous section, it will be worth developing a web application, which will lead the end user through the whole process of strategy creation. The core idea is to consult the team that made the first strategy and the current organisation board to develop a user-friendly and intuitive interface, which will provide the description of each analysis necessary for the process. As a consequence, one would not have to dive deep into any external resources and would be able to create this vital document from scratch by themselves.

1.3 Goal and scope of the thesis

The goal of the thesis is to prepare the operational strategy for ESN-EYE Lodz for years 2021-2023. The sub-goal is to develop an application, which will help the end user to follow the process of strategy creation. To achieve the both goals, firstly there was studied the current state of the art, which includes both programming and strategic management. The technological investigation contained mostly analysis of the frameworks and design patterns used for development of web applications. The managerial analysis focused on necessary steps during the strategy creation process.

1.4 Document structure

The document consists of seven sections. First is an introductory one. Next, there is described the current state of the art including the backend and frontend technologies comparison with argumentation for the best choice. What is more, there is also included description of tools used for strategy development process, as well as an overview of existing strategic management web applications. The third section contains information specific to the project, such as requirements, vision of the solution and UML diagrams. Section number four focuses on the most interesting parts of code and explains them in detail. It is followed by the user guide, which is a sort of documentation for the person that wants to use the application. Sixth section contains results of all analysis performed using the developed application. The final one sumarrizes the application creation process and indicates the features which could be improved or implemented in the future.

2 State of the art

In the following section there is reviewed the current state of the art including the business background, such as tools used for the strategy formulation. Later on there are discussed technologies, which could be used during the development process, as well as explanation, why the specific ones were chosen.

2.1 Strategy development tools

Strategic management is a core concept when it comes to managing any organisation. One can compare the organisation to a human being. Without a goal in life, person just lives day by day and, after some time, one can realize that it is just an existence, without any purpose. The same happens to the organisation, if it does not have an operational strategy. That is why, it is important to find out the long term objectives of the company, its mission and vision, as well as strengths and weaknesses. Fortunately, nowadays there is a significant amount of tools, which can be helpful during such process.
There is no set of tools that will always guarantee, that the strategy will be formulated perfectly for any organisation. There are couple of tools that should be used everytime like PEST and SWOT analysis or setting the long term objectives, but the rest is up to the one, who develops the strategy. In the next paragraphs there are described couple of tools, which are parts of the strategy development process.

2.1.1 Mission and Vision Statement

Every organisation in the world should formulate their mission and vision statement. These two describe the purpose of an enterprise's existence. To create the mission statement one should answers the fundamental questions, such as what does the organisation do, how it does it and why. The result should be formulated as a one simple sentence. On the other hand, vision informs the society what the organisation wants to become. While formulating such statement, it is worth asking, what are the hopes and dreams of the organisation and what it wants to change. As one can see, the main difference is that mission informs about present state of the organisation and its operational principles, while vision states how organisation would like to look like in the future [1].

2.1.2 PEST Analysis

PEST is an analysis of external factors which can affect the organisation [2]. The abbrevation informs about four types of factors, which should be included into such analysis:

- Political - this group incorporates all political and law issues. The source of factors can be for example environmental law, labour code, fiscal policy or the currently reigning political party programme.
- Economical - here one should pay attention to issues like uneployment, average salary of the citizens, inflation, tempo of economical growth or economical policy of the macroeconomy.
- Social - this group includes also cultural factors. Therefore, one should focus on demography, the level of education, religion, threat awareness, moral and ethical issues or society's habits.
- Technological - when determining technological factors it is worth thinking about current technology level, the focus on using new technologies, new discoveries and any other way in which technology can affect the macroeconomy.

Determining the above factors will allow to describe the macroeconomy of the organization and becomes extremely helpful during SWOT analysis.

2.1.3 SWOT Analysis

SWOT analysis is one of the fundamental tools used during strategy formulation. It includes four different factors that have to be analyzed:

- Strengths - these are the internal and positive factors. There should be listed any advantages of the organisation that make it better than the competition. For example if an organisation focuses on integration with international students, then a significant strength would be that the members know multiple languages.
- Weaknesses - this domain describes internal and negative factors. One needs to focus on finding out the potential weak sides, which affect for example the effectivity of the organisation. Things such as a frequent change of organisation members or their lack of knowledge are precise examples.

- Opportunities - the factors in this group are of external and positive type. Here, it is important to focus on which external factors can become opportunities for an organisation to develop or grow. This is the moment, where outcomes of PEST analysis become very useful.
- Threats - threats are external and negative factors. Similarly to opportunities, it is worth using the outcomes of PEST analysis here. However, this time one has to choose the factors, which can cause any trouble for organisation and which should be avoided if possible.

The properly done SWOT analysis allows to realize in which fields organisation should improve, which should be mantained and how external factors can be either useful or dangerous for the organisation's functioning [3].

2.1.4 SWOT matrix

After completing the steps in SWOT analysis, one can proceed with SWOT matrix. Firstly, one needs to create a table with all the factors from SWOT. Each strength and weakness should be put in a single row, while every opportunity and threat should be a separate column. The next step is to determine, how big influence each row has on each column. It is done by writing down the number in each cell of the tables. There are three possible level of influence values:

- 0 - no influence,
- 1 - low influence,
- 2 - high influence

The Figure 1 presents an exemplary SWOT matrix with all the data filled out.

		Opportunities		Threats	
		More International students at university	Experience of former members	Higher chances for students on the work market	Lower popularity of volunteering among students
Strengths	Innovative international projects	**2**	1	0	**2**
	Good recognition in ESN	1	0	0	0
Weaknesses	Frequent members rotation	1	0	**2**	**2**
	Big number of students with no experience	1	**2**	1	**2**

Figure 1: Exemplary SWOT matrix with ratings [Source: Own work]

As one can see, each cell contains value and thanks to that one can proceed with the next step, which is to pick the pairs with the high influence and write down the following solutions:

- For pairs strength-opportunity, one should find how strength can maximize the opportunity,
- For pairs strength-threat, one should find how strength can minimize the threat,
- For pairs weakness-opportunity, one should find how to minimize the weakness by taking advantage of the opportunity,
- For pairs weakness-threat, one should find how to minimize the weakness in order to avoid the threat.

These pairs, alongside with the ratings, let determine which type of strategy the organisation should stick to. There are four types available:

- Maxi-maxi strategy - it means that organisation has lots of strengths and external opportunities. Such strategy type allows to expand and grow, because both external and internal factors are favourable.
- Mini-maxi strategy - it means that organisation has significant amount of weaknesses, but lots of opportunities. It should focus then on taking advantage of opportunities and at the same time eliminating its drawbacks.
- Maxi-mini strategy - it means that organisation has lots of strengths, but external environment is not advantageous. Then the organisation needs to take the maximum out of its strengths in order to face the threats.
- Mini-mini strategy - it means that organisation not only has significant amount of weaknasses, but also very negative external environment. Such situation is pretty difficult as the organisation cannot develop, what leads to just trying to survive.

Basing on how many pairs of which factors were found out, one should pick the right strategy type for the organisation.

2.1.5 Long-term objectives

When the previous steps are done, one should focus on formulating the long-term objectives. They should be formulated in order to reflect the solutions found during the SWOT matrix analysis. Every objective should be stated according to a SMART rule [4], which means that any goal should be:

- Specific - it has to be very specific, so it could not be interpreted in many ways. When one sees the goal, it should be clear at the first glance, what organisation wants to achieve.
- Measurable - it is worth putting the numbers here, for example "gaining 10 000 more likes on social media" in order to be able to measure the progress.
- Attainable - if the goal is not real, there is high chance of losing the motivation to reach it.
- Relevant - the goal cannot be out of the organisation's business domain.

- Time-bound - it is important to put the time frame, in which the objective should be achieved, for example 3 years.

If the goals are formulated according to the principles above, it is way easier to monitor the progress and chances of meeting the expectations are significantly higher.

2.1.6 Balanced scorecard

In short, balanced scorecard is a tool that allows to measure the success of the strategy. It represents long term objectives as Key Performance Indicators (KPIs) with their target values [5] . Therefore, if a goal of the organisation is to get 50 new members each year, the KPI and target will be "new members" and "50 per year" respectively. All the measures shown on the balanced scorecard are grouped into four categories:

- Financial - the financial goals associated mostly with profit, revenues or operating costs of the organisation.
- Customer - the things that the target audience wants from the organisation and how is it seen by them.
- Internal processes - the services or products in which the organisation should be the best.
- Organisational capacity - the skills, culture and knowledge of the members.

The domains described above were designed for the enterprises' purposes. In terms of a non-profit student organisation the domains could be sligthly changed, as especially the financial goals would be difficult to define.
In the Figure 2 one can see the exemplary balanced scorecard done during one of the projects for the Royal Dutch Shell company.

	Goal	Objective	Measure	Target	Time
Financial	Developing competitive advantage	More efficient production of gasoline and diesel	Percentage change in efficiency from one barrel	Increase by 5%	1 year
	Increase overall revenue	Capital budgeting, only profitable projects, lowering risk of failure	Last year's statement comparison (net income, revenue, total and tangible assets	Increase all of them by 5%	2 years
Customer	Increase amount of biofuel available to customers	Introducing new technologies, efficient way of production, smart researchers.	Compare production with previous years	Increase production by 15%,	2 years
	Entering new markets	Focus on the introducing our products to new markets in Asia and Africa	Number of new countries in which we introduce our products	10 new countries	3 years
	Maintain loyalty of existing customers	bigger fuel discounts for card holders over 2 year, 50% discount on hot-dogs and coffee	Number of loyalty cards in use, percentage rise of hot dogs and coffee.	20% more cards in use, 30% more of hot dogs and coffee sold	1,5 year

Figure 2: Balanced scorecard for Royal Dutch Shell company [Source: Own work]

As one can see, the pretty vague goal is represented by specific objective, which has a specific measure followed by its target value and time frame. As mentioned before, this representation of the objectives allows to measure the effectivity of the strategy in a clear and concise way.

2.2 Existing applications

Strategic management or planning are wide terms. It means that there are already numerous applications, which focus on these areas. However, to determine what they are capable of, one needs to perfom and in-depth

analysis. To check the applications helping with strategic planning that already exist it is worth exploring the *Capterra* [6] website. It lists the best tools in such a category. It also allows to apply some filters and thanks to them, it was possible to shorten the range of applications just to free solutions, as the ones with subscription demand to contact a company's representative to schedule a demo. When it comes to the functionalities filters, the ones checked were "Goal setting/tracking", "KPI monitoring" and "Scorecards" as the resulting application would have similar features. The search outcomes were sorted in a descending order by the number of reviews. Each application was compared in terms of performing competitive analysis, setting the long term objectives and creating the scorecard with KPIs.

2.2.1 Monday.com

Monday.com [7] is a software developed by monday.com company. It has a high rating (4.6/5) alongside with about 2200 reviews, what makes it being considered a reliable platform. Browsing through the features, there is a huge variety of them and, what is more, all of the mentioned in the introductory part are included. The application is based on the templates for each feature. At the home screen when there is a workspace added, it redirects user to the template center. One can pick the one that is suitable for their case then. The interface is shown in the Figure 3.

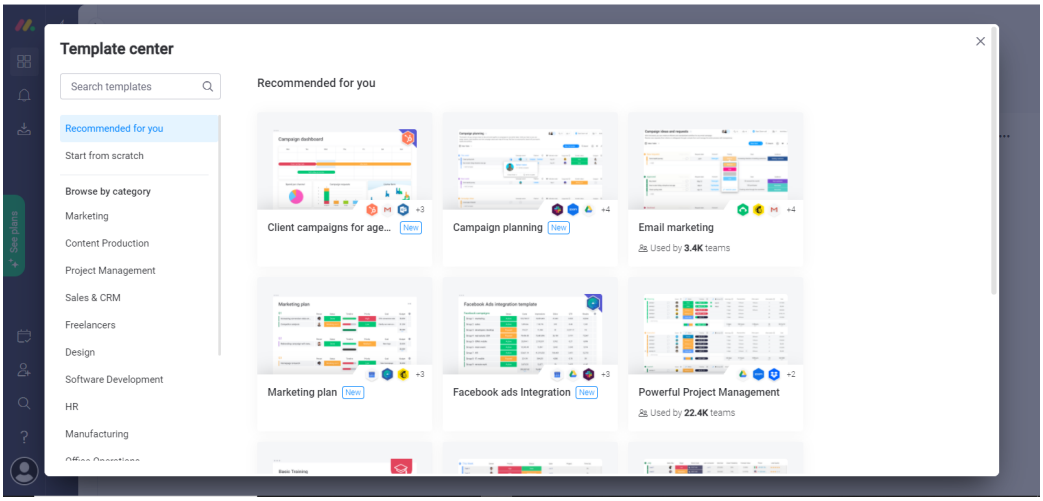


Figure 3: Template center screen [Source: Monday.com]

Starting with the competitive analysis, one can choose the *competitors analysis* template. Figure 4 represents how it looks like.

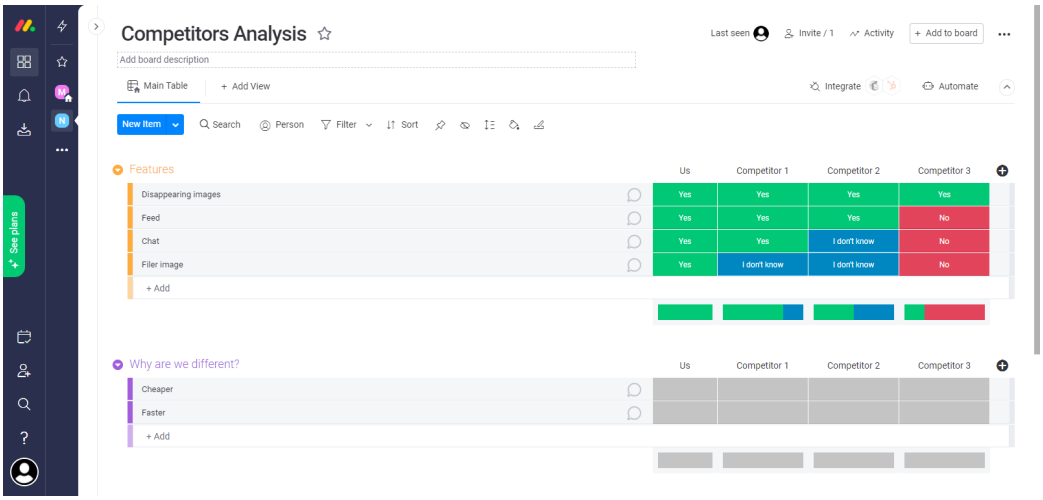


Figure 4: Competitors analysis template [Source: Monday.com]

As one can see, it is divided into three sections. First one lists the features and allows to determine, whether the competitors and our company

are capable of providing some services. The second one presents why our company is different than others, for example in terms of services speed delivery. The last one focuses only about sales of the services, including the pricing plans. These three sections can give a brief overview on the advantages and disadvantages of our company and the competitors. It looks similar to the SWOT matrix mentioned before, but is focusing less on our company. On the contrary it gives a great overview on the competition on the market.

When it comes to the KPIs and scorecards, there are no specific templates for that, but the great advantage of the software is that any template can be easily adjusted. In this case, the *Project management* template has been modified to suit the needs of KPI monitoring.

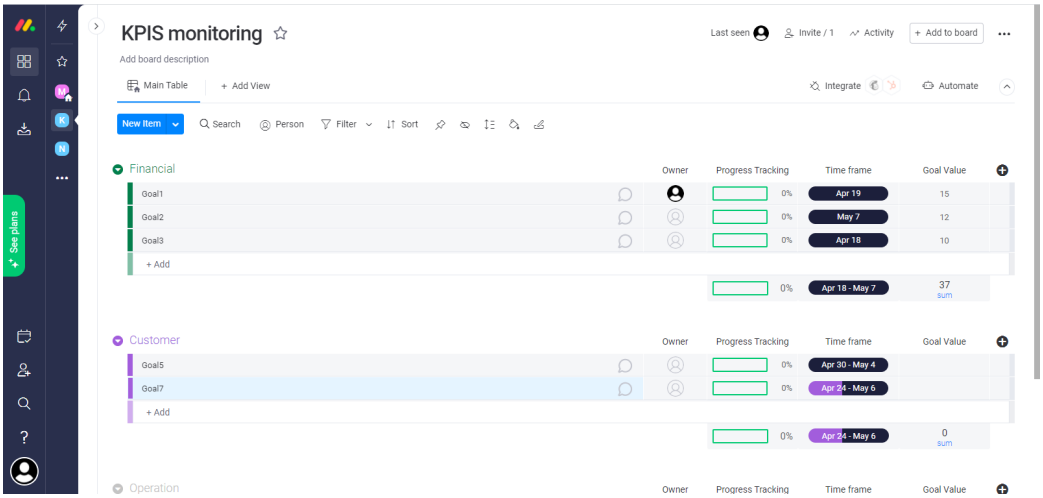


Figure 5: Template adjusted for the needs of scorecard and KPI monitoring [Source: Monday.com]

As one can see in the Figure 5, all the necessary data for KPIs is visible and they are grouped just like in a balanced scorecard. Progress tracking column allows to be updated about the progress all the time. If someone needs, there can be added any other measures in other columns. However, there are two small drawbacks. Firstly, on the time frame preview one cannot display a year, what in terms of strategy would be way more useful. The second one is that the goal value can just be the plain number, so multiple

rows cannot have different units, like percentage or currencies.
The general impression on the application is great. It is free for personal use and has a huge variety of different templates. The interface at first glance seemed quire complicated, but after adding the first template everything cleared out. Unfortunately, the tool does not contain the templates useful for PEST or SWOT analysis, so they would have to be done by the user from scratch. Anyway, it is totally understandable, why the tool has that high rating on *Capterra*.

2.2.2 Jedox

Jedox [8] is a software developed by Jedox company. It has a high rating (4.4/5) with about 100 reviews, therefore it is difficult to determine at the first glance if tool is a suitable one. On the other hand, when one logs in, the user interface looks way more confusing than the one on monday.com, what is shown in the Figure 6.

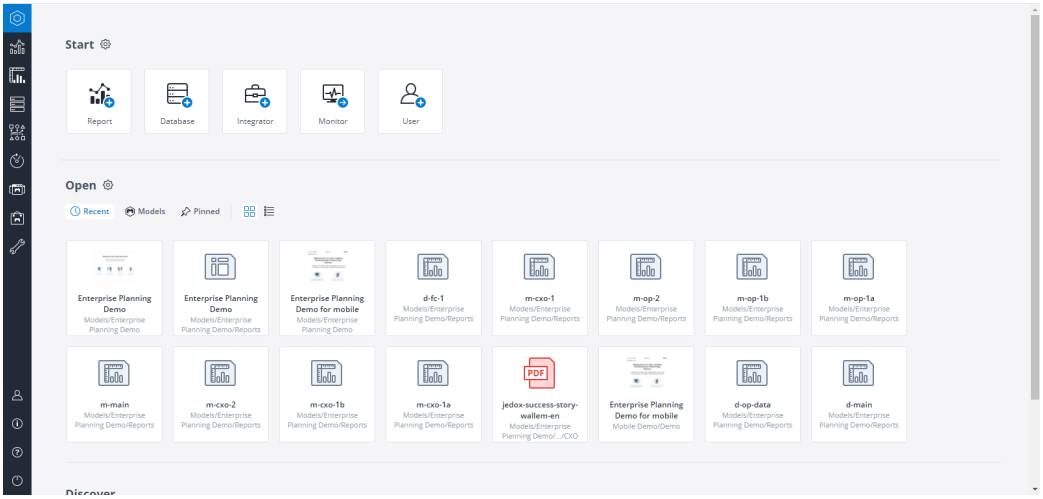


Figure 6: Jedox application homepage [Source: Jedox]

It is difficult to find the functionalities that were listed on the [**?**] page. When entering the example of planning models, there are no visual representations and it is definitely difficult to find out what is going on. The tutorials are also very vague. In general, the application seems more a tool for enterprise

budget management and sales statistics, as it contains excel spreadsheets plugin and other financial tools. Using it for strategic planning seems totally useless.

2.2.3 Wrike

Wrike [9] is a software developed by Wrike Inc. It has a high rating (4.2/5) with about 1600 reviews, what shows that it is a popular and reliable application. When one logs in, one can see the screen like in the Figure 7 .

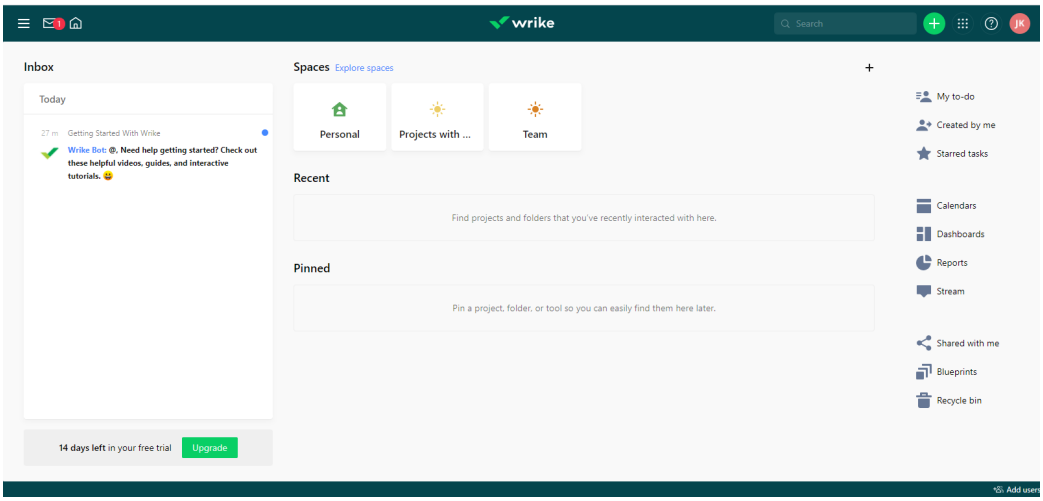


Figure 7: Wrike application homepage [Source: Wrike]

At a first glance, the user interface looks concise and clear. It is easy to find out, which icon leads where and what are the basic components of the application. The helpful issue is that when one accesses the application for the first time, there is a video tutorial provided. In case of Wrike it is very descriptive and clear, like it was for monday.com. The main feature here seems to be the tasks board, which consists of the list, board and table views, as well as a Gannt chart. It can be helpful during the strategy development, but just in terms of organising the work and deadlines. If one would like to create any analysis needed for the strategy creation, it would have been done by themselves. Therefore, the tool does not seem a good solution for such task.

2.2.4 ESM+Strategy

ESM+Strategy [10] is a software developed by ESM company. It has the highest rating (4.7/5) among the others, but the review number is equal just to 32, what cannot determine whether the tool is really the best. However, out of the other solutions stands out the fact, that on the pricing page there are mentioned couple of tools, which were not seen anywhere else, such as SWOT and PEST analysis. One can see that in the Figure 8.

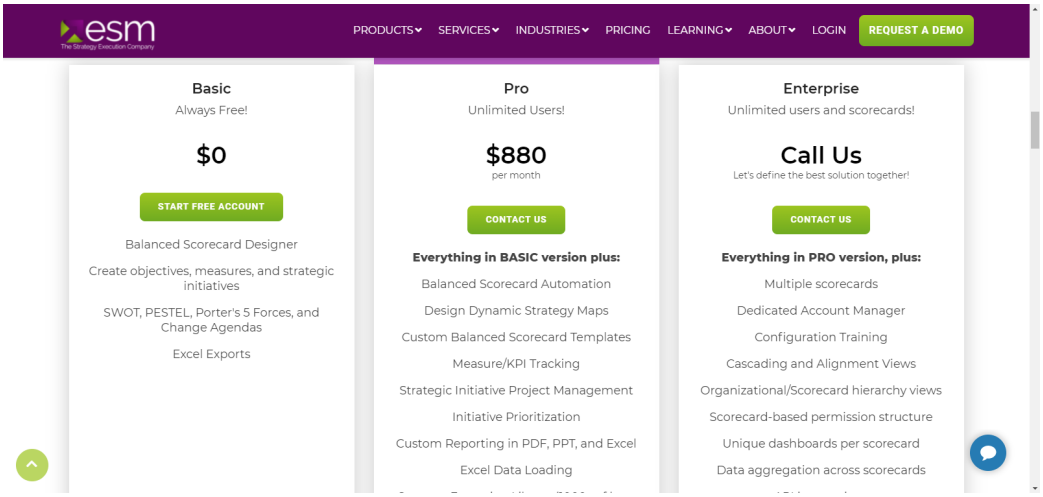


Figure 8: ESM pricing page [Source: ESM]

The Figure 8 shows that there is high chance of developing a proper strategy even with basic tools. When one logs into the application, the interface looks concise and intuitive, what is shown in the Figure 9.

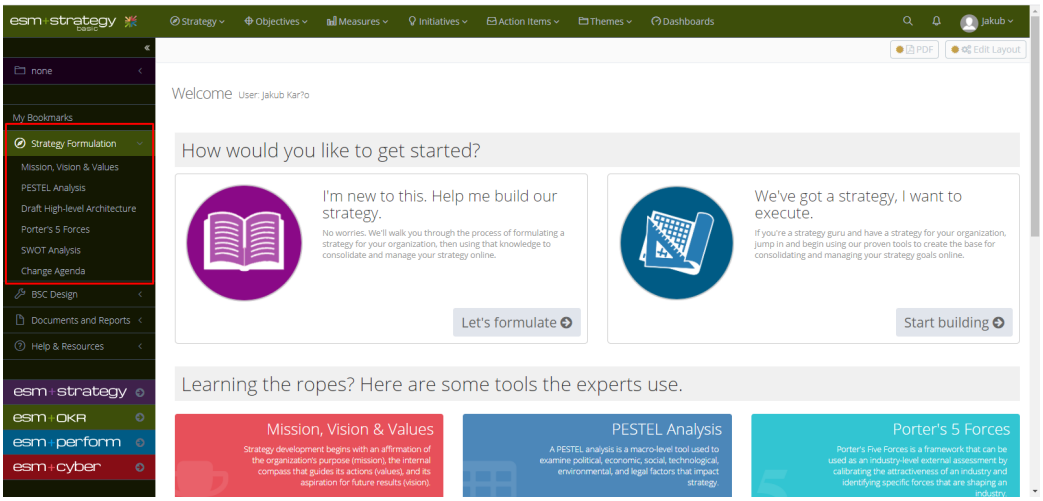


Figure 9: ESM+Strategy application homepage [Source: ESM]

As one can see, on the left panel there are available the tools such as mission and vision statement, PESTEL and SWOT analysis or Porter's Five Forces analysis, which are tools often used during the strategy formulation process. On the example of PESTEL one can confirm that interface is user-friendly and simple. It also contains the help tabs, in which each step is described. The Figure 10 presents the PESTEL analysis page.

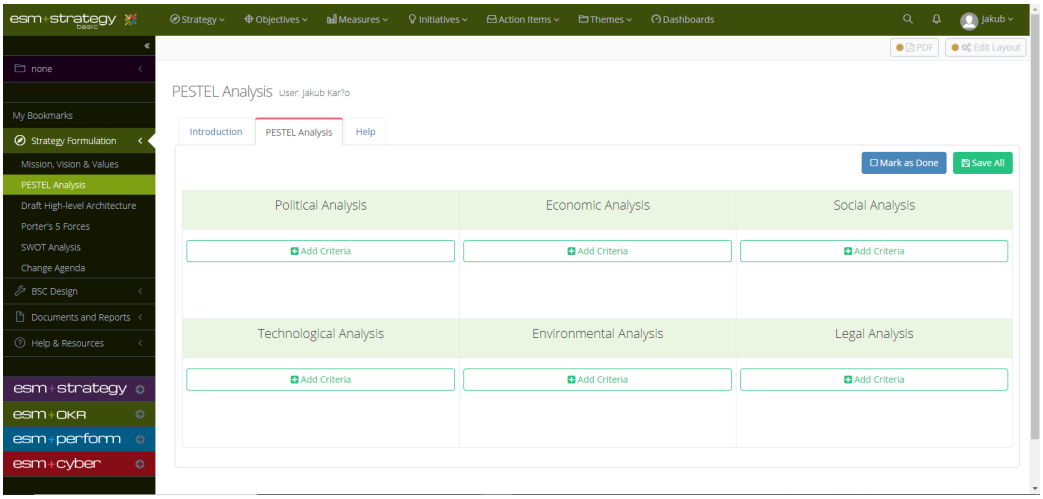


Figure 10: ESM+Strategy PESTEL analysis page [Source: ESM]

The SWOT analysis tab looks pretty similar to PESTEL but the thing that it lacks, is the list of external factors from the previous analysis displayed somewhere nearby, in order to have an eye on the potential opportunities and threats, which can affect the organisation. It is shown in the Figure 11.

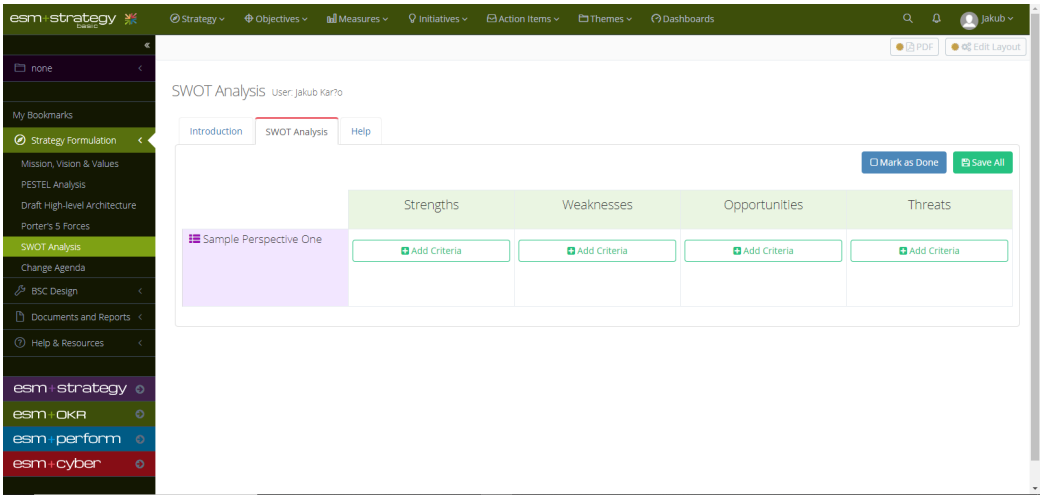


Figure 11: ESM+Strategy SWOT analysis page [Source: ESM]

Unfortunately, the free plan does not allow to automatize the balanced scorecard generation, so all the data have to be input from scratch. It would be really nice to try the solution, which makes the scorecard by itself. However, there is no way to try it out without paying for the upgraded version.

In general, the application is great for strategic planning purposes. It contains essential tools even in the free plan and like Wrike or monday.com it has also an intuitive UI. There is no tutorial at the beginning, but it is not needed.

2.2.5 Conclusion

It is very difficult to find an application that is free and contains the necessary functionalities. Most of the applications demand to get a demo and they are premium versions. Anyway, it was possible to find interesting solutions with lots of useful tools. In general, despite mentioning strategic planning as a feature, most of the applications seem to be the ones, which help to execute the strategy, but they do not consider generation of such documents by itself. Only ESM+Strategy was the one, which was designed especially for strategic planning purposes. The basic version is useful, but it would be great, if it could have couple more options, such as display of previous analysis results while working on next one. In addition, it still does not reflect all the steps of that would be executed during strategy development for ESN-EYE. The comparison shows that the idea of an application, which is only used for developing the strategy seems attractive and necessary, as only one of the applications was able to help with strategic management itself. What is more, ESM application is designed for business purposes, while students organisations have slightly different needs. As the motivation for writing the application is to help Erasmus Student Network EYE in a significant way, it means that creating new one specifically for the organisation is a necessary solution.

2.3 Server-side and client-side applications

When it comes to web application frameworks, developer has various options to choose from. Therefore, it is extremely important to dive deep into the technological scope and determine, which technology will be the most beneficial for the project. Firstly, it is crucial to understand the difference

between server-side and client-side applications.
In general, the internet works on a client-server model. It means that everywhere in the web there are located servers with specific IP addresses, to which the users (clients) connect through the network to obtain any data or information they need. The client's device sends a request to the server and gets a response. That is the core of network communication. Therefore, one can distinguish the client-side and server-side applicatons. The first ones run on a client's machine mostly through web browsers, while the latter run directly on the server.
For the purpose of the application, there will have to be developed two parts. One of them the server-side application which communicates with database and manages the business logic. The second one is the webpage that will display all the necessary data in the client's browser. As a consequence, the following analysis of the technologies will dive deep into couple of available web frameworks in order to determine, which one will be the best to use.

2.4 Backend technologies

2.4.1 Enterprise Java Beans - EJB

Enterprise Java Beans is a Java API used to program the component-based server-side applications [11]. It defines a set of services that are provided by the application container. The container knows which dependencies it should use thanks to the annotations in the Java code. For the view, EJB uses JavaServer Pages. It is an engine, which changes the webpage code into the servlet. To run the EJB application, one needs an application server, such as JBoss or Weblogic. The main problem with EJB is that the size of application does not matter - the configuration is done the same way both for big and small projects. As a consequence, there is high probability that application package will contain redundant libraries and code. What is more, it is nowadays obsolete as there was introduced Spring framework, which has already become a standard when developing Enterpise Java applications [12]. EJB is used only by about 7% of developers according to JRebel report from 2020 [13]. Therefore, EJB would be one of great options to develop the backend services and, what is more, it allows to prepare the frontend in one application too. However, it needs a lot of configuration and, additionally, it would be worth developing the application sticking to the more popular and modern software development practices, as well as the architectural

standards.

2.4.2 Spring

Spring is an application framework written in Java, which allows to develop the server-side applications [14]. It became a reliable alternative for EJB framework, since the programmer does not have to get all dependencies of the framework at once - any of them can be added anytime. It makes development easier and faster, as unnecessary dependencies are not contained in the application package. It comes with a template engine, Thymeleaf, which allows to render the data from the backend application directly to the view.

Spring is a perfect framework for developing the service based applications. In the Figure 12 one can find an exemplary microservice schema.

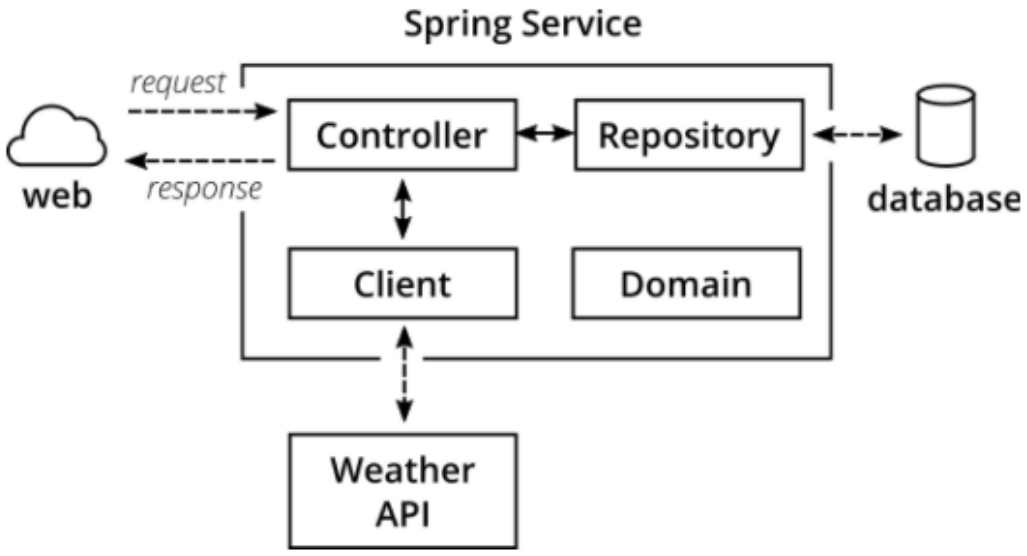


Figure 12: Spring Service structure [Source: Pinterest [15]]

In the Figure 12, there is an external web application, which represents a View layer. It sends requests to the backend server created using Spring Framework. The request is handled by the Controller layer, which redirects it to the responsible class. If it is a database operation, the request is passed to

the repository, while if one needs to for example get data from another API, it is done via a client service. The domain class type is a part of the Model layer. It represents either a database entity, thanks to Object-Relational Mapping, or a POJO, which stands for Plain Old Java Object. The response is returned to the View and displayed to the end user.

2.4.3 Spring Boot

Spring Boot is a project based on Spring framework. It allows to bootstrap the Spring applications without producing a lot of boilerplate code. Whenever one creates a project, it contains the dependencies that are necessary to run. What is more, it comes with an embedded server, which makes the deployment process easier. It wraps up couple of other Spring projects, such as Spring Data, Spring Security or Thymeleaf, a template engine, which will be described in the next section.
According to JetBrains, the one of the biggest players on the development tools market, in 2020 Spring Boot was used by 63% [16] of Java developers for web development. The second place is taken by Spring MVC with 43% [16], what shows that knowledge of Spring and Spring Boot is right now inevitable in the world of Java programming. What is more, the framework is very well documented and supported.

2.4.4 Framework choice

Considering EJB, the framework is very powerful, but it is becoming obsolete and demands a lot of configuration. Spring Boot seems to be the most optimal framework thanks to Thymeleaf, as the view, which will be sent to the user, can be embedded directly into the backend application, what saves a lot of time, because there is no need to use another framework. The only doubt could be that in general view rendered on a server side is a slower solution than doing it on a client side, especially when one wants to make dynamic reloading. The reason is that on the server side the whole HTML document is rendered every time one wants to any element dynamically, while on the client side JavaScript allows to load just the small chunks of the HTML code [17]. It could be a problem mostly when lots of users try to access the server. Fortunately, the resulting application will be accessed maximum by 5 people at once, since only ESN-EYE Board members will be able to use the system. Consequently, Spring Boot seems to be perfect

framework for the implementation process.

In the next part the area of focus will be the necessary technologies during Spring Boot software development, such as the database implementation, programming language and servlet container.

2.4.5 Java

Spring Boot development language is Java. It is an object oriented programming langu,ciple. It means "Write once, run anywhere", what in different words means that it is platform independent, so the compiled Java programme will run identically on Unix, Macintosh, Windows or any other operating system [19]. This is possible thanks to Java Virtual Machine (JVM), which is an abstract computing machine through which every Java programme is executed. When source code is compiled into the Java bytecode, the operating system specific JVM allows to execute it [19]. This means that every operating system needs different version of JVM, but the code is compiled according to the same rules.
Currently, Oracle, the owner of Java, do not support all of Java Development Kit (JDK) versions. The last one, that is totally free and with Oracle support is Java 8, released in March 2014. For official support for the next versions, there is a subscription needed. However, since September 2017 Oracle provides JDK releases under a free open source license [20] . It is called an OpenJDK and it is being supported by the developers community. As a consequence, it is possible for a developer to use the newest JDK versions for free, but in case of errors in software, they have to ask the community for help instead of the Oracle company.

2.4.6 Relational databases

Relational databases are based on Structured Query Language (SQL). The name comes from tables that have relations between each other. For a long time they were actually an only option of providing the database services. Each table has a set of columns, which determine the specific properties of the database entry. Every entry is called a record and any table can contain a huge amount of such. The greatest advantage of relational databases is that they provide better data consistency than non-relational ones [21]. It is called an ACID principle, which states that for any transaction, the atomicity, consistency, isolation and durability will be kept. It guarantees that no data

will be lost. Therefore, if the integrity and reliability of the data transactions is the key point, it is worth thinking about using the relational database implementations, such as MySQL, OracleDB or PostgreSQL. What one can also consider one of the pros, it uses a standarized query language, which is only slightly adjusted within different systems, so if one already knows SQL, there should be no problem with developing the structures in any of the systems mentioned. On the other hand, any changes in the structure of the SQL based systems is time consuming, as one has to change the existing table structure. As a consequence, if one wants to use such solutions, the whole data structures need to be thoroughly analysed beforehand.

2.4.7 Non-relational databases

Non-relational databases are also called NoSQL systems. They can store data in couple of ways, including JSON-like documents, key-value pairs or graphs containing nodes and edges. This is a different approach to relational databases, which use tables for data storage. In addition, this system uses dynamic schemas, which allow to create entities without defining their structure first [22]. In relational systems, the structure has to be defined primarily, otherwise there is no option of creating data records. The dynamic approach became very useful during application development process, as some structures needed to be modified due to primary assumptions modifications. Such systems are used widely in the applications that constantly grow and demand frequent changes in the data structures as well as process huge datasets. The examples of such databases are MongoDB, Cassandra, ElasticSearch or HBase [23]. One of the drawbacks is that if the system is not perfectly implemented, there is a chance of data loss, because it lacks the transaction support management. What is more, the non-uniformity of NoSQL databases can also be confusing, as if one needs to switch to any other implementation, it is highly probable that all the structures will have to be reorganised.

As the application will have a structure that is not about to dynamically change and would not store huge amounts of data, choosing a relational database sounds a better option. In addition, Spring Data has great support for Object-Relational Mapping for such databases. Furthermore, even if the amount of data would not be huge, sticking to ACID principle for each transaction is another key point. When it comes to specific database system, another important criteria is the free access, what implies that OracleSQL

needs to be neglected as it is a paid software. Therefore, one should choose between MySQL and PostgreSQL. Configuration of both in Spring Boot is not a demanding task. The difference that could become the key, seems to be that PostgreSQL is actually object-relational database management system rather (ORDBMS) than just relational (RDBMS) one like MySQL [24]. For a person that is familiar with object-oriented programming languages, such as Java, it can become handy, as for example tables can inherit properties from each other.

2.4.8 Apache Tomcat

When Spring Boot application is compiled and ready to run, it needs to be uploaded into a servlet container. The most popular one is Apache Tomcat. When it comes to deployment using such solution, there are two options to choose from. It can either be an embedded Tomcat that is provided with Spring Boot, or a standalone instance. The second solution does not require configuration of the container for each application, as there just need to be uploaded Web Application Resource (war) files, which can be hosted by a single Tomcat instance. Another options for deployment are for example JBoss and or Weblogic, but the latter one is a paid solution and if Spring Boot already is ready to deploy to Tomcat, then it seems to be the best choice.

2.5 Frontend technologies

After choosing the framework for backend development, one should review the most popular frontend frameworks.

2.5.1 AngularJS

AngularJS is a client-side framework built with TypeScript. It includes a collection of libraries, which allow to change the static HTML code into dynamic components. It can be built using NodeJS [25]. The configuration is quite simple, what allows to start the development fast. What is important, it is maintained by Google, a big player on the software development market, what means that there is high chance that the framework will be constantly improved and will remain stable [26]. It is great for dynamic single-page applications, however it still needs an application written in different framework

on the server, as there is no way to develop the backend application with Angular.

2.5.2 React.js

React.js is another client-side framework. It is a JavaScript library which allows to build the user interface. It is the most popular framework on GitHub, as there are about 136 000 projects [27], when one checks the tag "react". It is owned by Facebook [28], so after Google and Angular, there is another big player involved. What is great, the React code can be used to develop not only web applications, but also mobile and desktop ones. Unfortunately, for the purpose of the thesis, React is also the client-side only, so it will demand from the developer to create two apps - frontend using React and backend with another framework.

2.5.3 Thymeleaf

Thymeleaf is a Java server-side template engine. It is able to not only process HTML code, but also XML, JavaScript, CSS and, in addition, plain text. It takes advantage on of the so called processors, which are nothing else than objects, which add the business logic to the markup tags. The set of such processors is called a dialect. The strength of Thymeleaf is that the standard dialect provided by default can be extended in any way. Spring developers decided to do it and created SpringStandard Dialect, which is almost the same as the basic one, but with slight adjustments for Spring environment [29]. In comparison to JSP, which has its specific markup tags, Thymeleaf uses the standard HTML tags with adding some attributes. This makes the template more reusable, as the only thing bound to the code are properties of the tag, not the tags themselves. Thymeleaf is widely used as for past year the downloads number from Maven Central repository oscillated between 800 000 and 1 000 000 per month, what can be seen in the Figure 13.

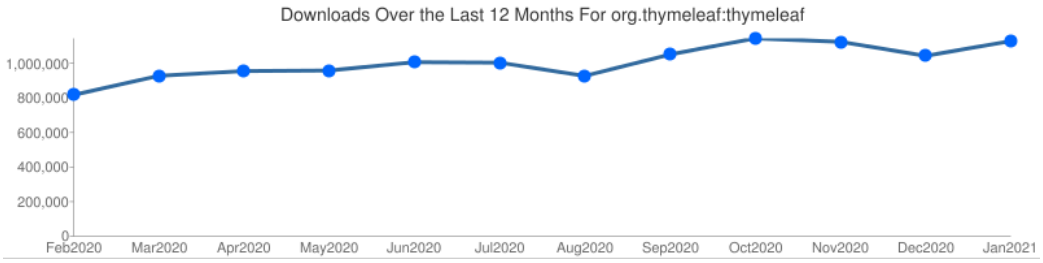


Figure 13: Downloads number of Thymeleaf from Maven Central repository between February 2020 and January 2021 [Source: Thymeleaf Official Webpage [30]]

2.5.4 Framework choice

Both AngularJS and React.js are powerful frameworks, but they are only client-side, what means that there will have to be developed also the backend service. Since Thymeleaf can be developed in the same project as the backend, thanks to Spring Boot and, what is more, is supporting Java, in which the server side of the application will be written, it seems better choice than any of the frameworks based on JavaScript.

2.6 Summary

Basing on the analysis of both backend and frontend technologies, the decision was to develop the application using Spring Boot alongside with Thymeleaf as a template engine. The database system chosen was PostgreSQL.

3 Project description

The following section describes the project in a specific, detailed way. The first part focuses on the requirements, as well as vision of the entire solution and the domain model. In the second one there are presented the diagrams, which will help to understand the design principles of the application.

3.1 Requirements identification

The application needs to fulfil couple of requirements. Functional requirements are responsible for describing functions of the system, while non - functional describe the limitations or restrictions upon which the system works.

3.1.1 Functional

- Authentication - The user can log in to the application via Google Oauth2 by using their Google account.
- Strategy initiation - To initiate the strategy, the user needs to provide a title.
- Vision and Mission statement formation - The user has to provide two strings - one for the mission statement and second one for vision statement. They can be edited anytime.
- PEST analysis - The user can provide numerous amount of external factors affecting the organisation.They are divided into four types: political, environmental, social and technological. Each type has its own input window. User can save the current state of the analysis anytime.
- SWOT analysis - To conduct SWOT analysis, user has to complete PEST analysis first. The user can provide the elements of the analysis divided into four types: strengths, weaknesses, opportunities and threats. It is possible to take advantage of factors done in PEST analysis. Each type has its own input window. User can save the current state of the analysis anytime.
- SWOT matrix - It can be done only after conducting SWOT analysis. The user can estimate the affection level between each two factors.

There are three levels to choose from: none, low and high. User can save the current state anytime.

- Long-term objectives - It can be done only after finishing the SWOT matrix. The user has to provide the name of the objective, its description and time period in which it should be achieved. The user can take advantage of previously conducted analysis. User can save the current state anytime.
- Document generation - It can be only executed, when all the steps of strategy development are done. User can generate the outline of the strategy document divided into sections with basic descriptions. The descriptive part will have to be done by the user itself.

3.1.2 Non-functional

- Only ESN-EYE Board members can log in into the application.
- All users can see all created strategies.
- During each step of strategy development, user will be able to check the definitions from the field of strategic management in order to formulate statements and conduct all the analysis properly.
- To conduct the next step, the previous one needs to be completed.
- Removing the strategy removes all associated analysis as well.
- The generated file is not a complete strategy.
- The SWOT analysis factors can be created from PEST factors.
- Each Long-term objective has to contain the time period.

3.2 Solution vision

As mentioned in the section 1.2, the core idea is that the user without any experience in strategic management domain will be able to create the draft of the strategy by themselves. Even with help of the application, the process demands a lot of input from the user. It implies that the interface must be simple as well as contain a lot of hints connected with the strategic

management concepts. In the following sections there is described the idea behind backend API and frontend implementation.

3.2.1 API

The idea was that there would be developed a REST API using Spring Boot. The analysis of the possible solutions determined that the core of the API should be just Create, Retrieve, Update and Delete (CRUD) operations performed on the database. For each existing entity there should be implemented a repository layer, which would allow the flow of the data. The additional feature is the .docx file generation, which contains all the data from the application which are necessary for such document.
In terms of security, each user should be authenticated. Thanks to the fact that all ESN-EYE board member accounts are registered on Google domain, then the natural choice was to choose the Google Oauth2 authentication provider. New board members always are confused at the beginning due to lots of access data for different platforms, so creating a new account just to use this specific application would be pointless.

3.2.2 Frontend

For the end user, the frontend part is the most important one. No one would use the most complex and powerful API, if the user interface and user experience is poor. Therefore, all data obtained from the backend had to be represented in a consistent, readable and intuitive way. The core idea for the view layer was to display all hints which user needs during the strategy development process. It would be also helpful if one could hide the hints for some time and come back to them whenever needed. Consequently, there was born idea of a sidebar with theoretical parts, which can be shown and hidden anytime. What is more, the colours of the interface should not be chosen just based on the intuition. After there was found the main colour of the interface, there was used the online tool, which allowed to pick the triadic colors, what is shown in the Figure 14 .

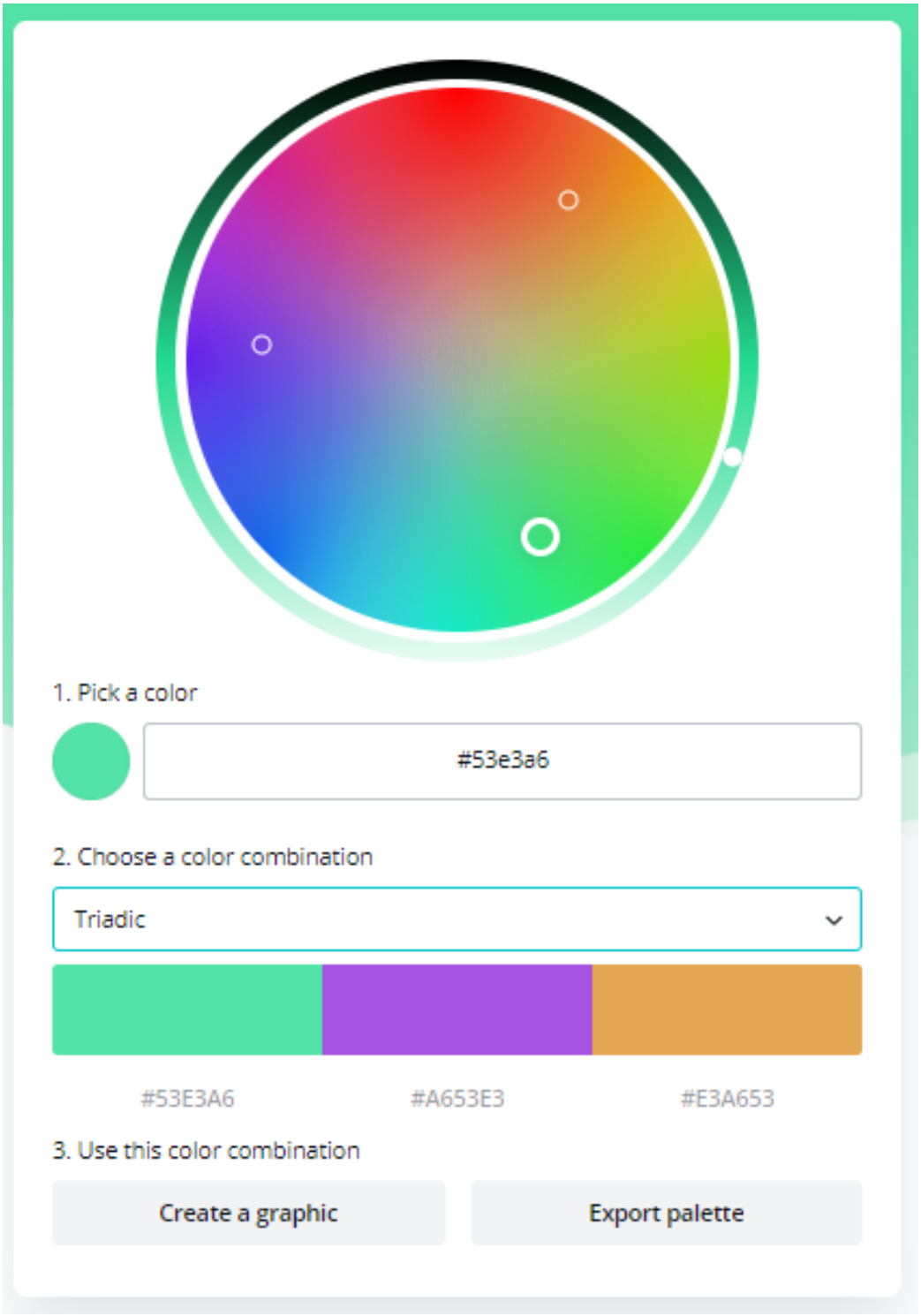


Figure 14: Tradic color schema [Source: Canva.com [31]]

Triadic colours provide the harmony and visual contrast [32], what can help to make the user interface eye-catchy. In addition, one could add a shade of black, which is one of the universal colours. Consequently, the resulting three basic colours of the application can be seen in the Figure 15.

Figure 15: Application basic colours [Source: Webfx.com [33]]

3.3 Domain model

The analysis of how domain model should look like focused on storing the entities in the database, as the API focuses mostly on CRUD operations. Since the relational database is used, the entities structure should be thoroughly considered before the implementation. Thanks to the Spring support on database operations, the mapping of the database entities to Java classes is relatively simple. In the Figure 16 there is presented a domain model schema.

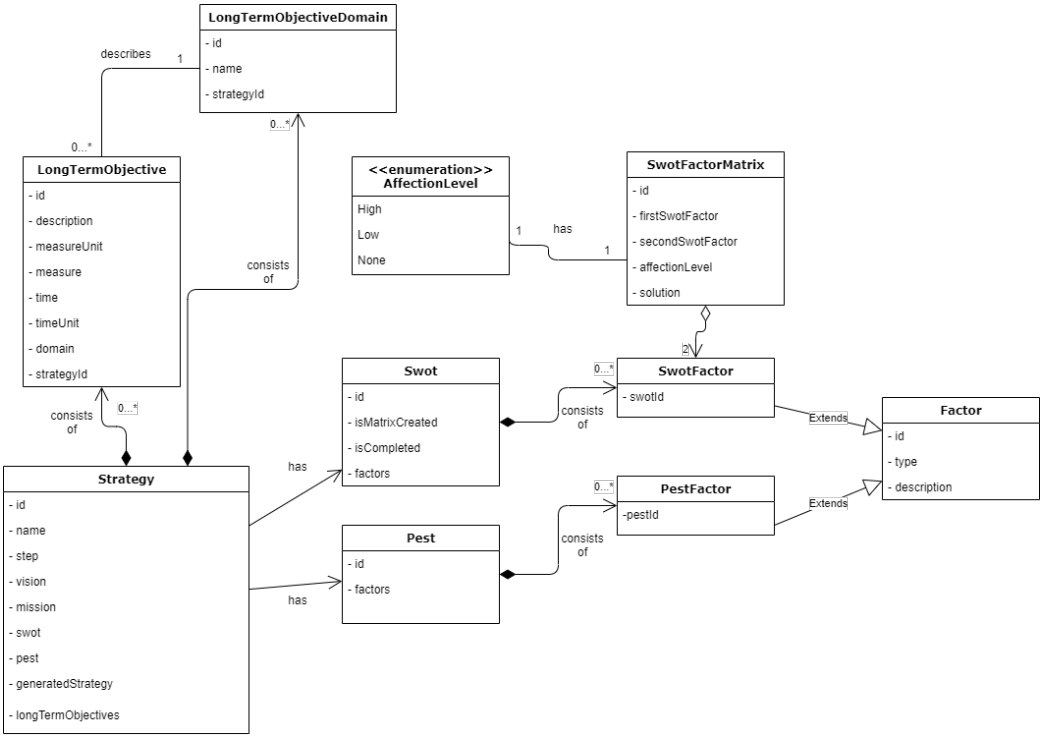


Figure 16: Domain model [Source: Own work]

The most important object in the application is *Strategy*. All other entities are based on it. As one can see from the model, both *Swot* and *Pest* are possesed by *Strategy* object, while each instance of *LongTermObjective* and *LongTermObjectiveDomain* contain the reference to it. Then there are lists of factors specific for both of the analysis entities and since their only difference is the factor type, which is anyway just a letter, and a reference to different entity, the inheritance of the properties from the parent *Factor* object became useful. Each *SwotFactorMatrix* instance contains reference to exactly two *SwotFactor* instances and it can be removed without affecting the other entity, therefore relationship between these two objects is an aggregation. The relation between *LongTermObjective* and *LongTermObjectiveDomain* is an association with multiplicity, since each object of the first type can contain only one reference to the latter one, but there can be a lot of objects which posses the same instance of *LongTermObjectiveDomain* type.

3.4 Design

3.4.1 Class diagram

In the Figure 17, one can find class diagram of the application.

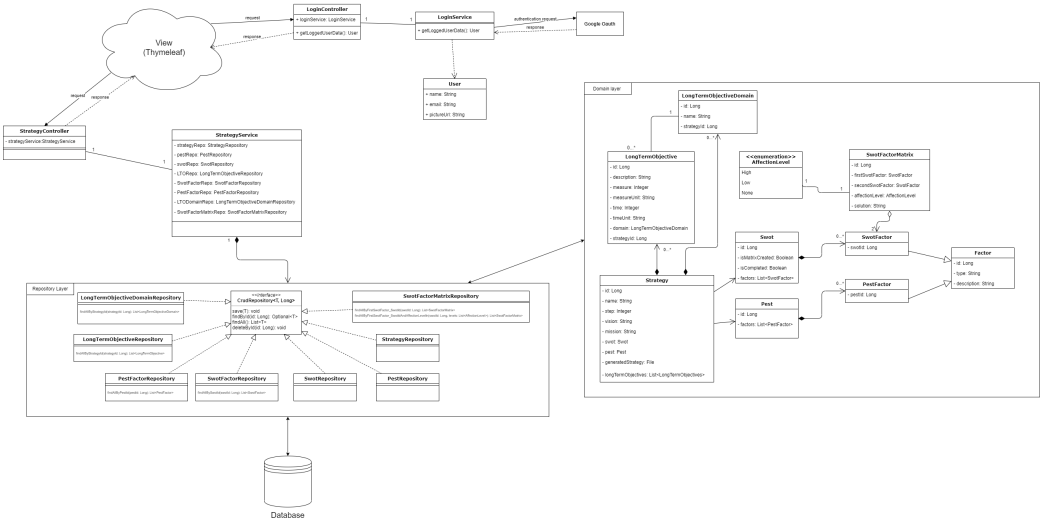


Figure 17: Class diagram of the application [Source: Own work]

The domain layer of the class diagram was thorougly described in the Section 3.3. As one can see from the diagram, each entity except *AffectionLevel* class possesses the corresponding Repository in the Repository layer. They are responsible for all the database operations, such as CRUD. All of them inherit the methods from *CrudRepository* interface provided by Hibernate. The methods of this interface, such as *findById()*, *findAllById()* allow to retrieve the data or create/update the object like *save()* method. However, some of them, like *SwotFactorMatrixRepository* contain the methods which allow to get the entity by specific database column, like in this case *swotId*. There are two classes, which are essential for the application - *StrategyController* and *StrategyService*. The controller redirects all the requests from the view developed in Thymeleaf and passes it to the specific methods in the service, as well as returns all the data to the view. At the same time, the service encapsulates all business logic of the application, such as manipulation of the data obtained from the user through controller or the entities retrieved from

the repository layer. As one can see, both classes contain a huge amount of methods, what makes the class contain many lines of code. However, since actually all the data from repositories can be accessed through one *Strategy* object and all other entities are a part of strategy development process, the decision was just to add multiple repositories to the *StrategyService* class. The same approach comes to *StrategyController* since all data passed through the service is also focused around one entity.
When it comes to the user authentication, it was mentioned before that the users already have Google accounts. Therefore, there was developed a *LoginService* which communicates with Google Oauth service and retrieves the basic user data, containing *name*, *email* and *pictureUrl*, which are returned to the *LoginController* in order to display the welcome screen for the logged in user.

3.4.2 Sequence diagram

The application was created to lead the user throught the one complex process. However, the core parts are two analysis: PEST and SWOT. For both of them from the application point of view, the process looks exactly the same, there are just different types of factors. Therefore, in the Figure 18 there is presented a sequence diagram of the PEST analysis development.

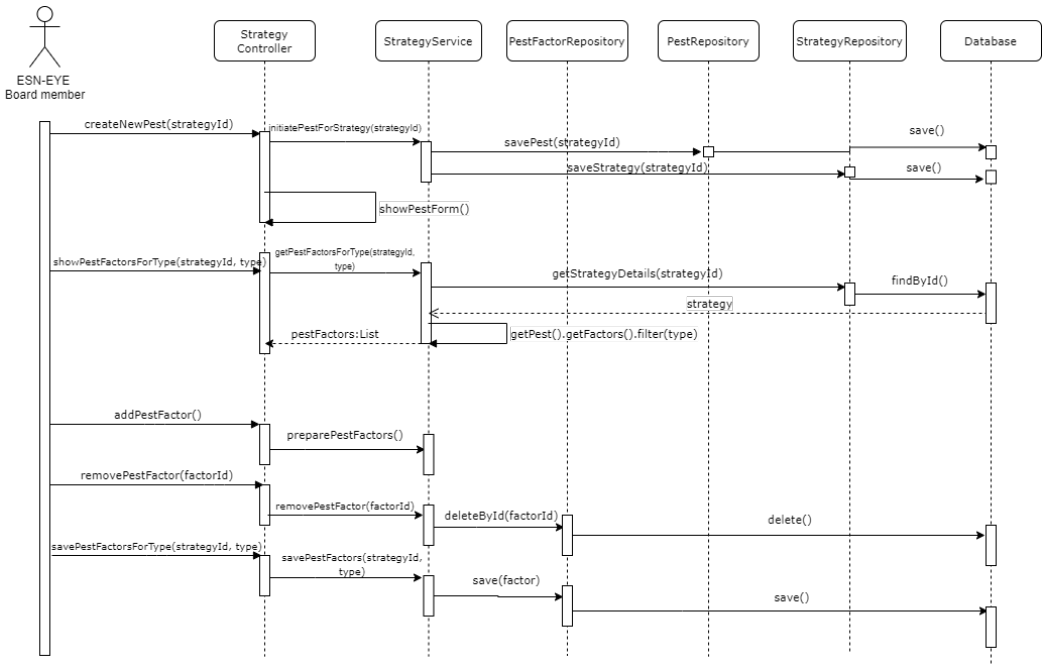


Figure 18: PEST Analysis sequence diagram [Source: Own work]

As one can see, all the actions are shown for the ESN-EYE board member. There are shown three important layers including controller, service and repository. The *StrategyController* and *StrategyService* classes are the core, which redirects the requests to the right repositories depending on the entity type used.

First, the user needs to create a new PEST analysis using the *createNewPest()* method, which initiates the entity in the database and assigns it to the right strategy. Right after, there is performed *showPestForm()* method, which renders the form in the browser. Then the user chooses, which factors they want to focus on. When they choose the type, there is performed a method *showPestFactorsForType()*, which calls for the strategy object from the*StrategyRepository* and through that object it is possible to access the PEST analysis and its factors. All the factors are filtered by the chosen type and displayed to the user as an editable list. Afterwards, the user is able to add and remove the factors followed by saving them. The *addPestFactor()* method calls the *preparePestFactors()* method, which checks if there already exist any factor of such type, and if not, there is created a temporary

list, which prevents the user from obtaining the NullPointerException while adding items to an empty list. What is crucial, adding process do not access the database before saving the whole list. On the other hand, when it comes to the removal process there has to be performed some database action, since just saving the whole list at the end would not remove the unnecessary table rows. Consequently, there is called a *removePestFactor()* method with *factorId* as an argument, which checks if there exists any table row with such id and deletes it if necessary. Finally, one can save the whole list of factors with specific type thanks to *savePestFactorsForType()* method, which calls *savePestFactors()*. In this method, there is performed a factor type check based on the view parameter, which lets to map the *String* type value into a database column, which just stores one character.

4 Implementation

In the following section one can find the implementation specification of the most iconic features.

4.1 Strategy generation

The outcome of the whole strategy process is the document in the **.docx** format, which will let the end user take advantage of all analysis that were performed in the application. After that it would be possible to write the parts of the document that were too descriptive to be provided by the application, such as the explanation why the document was created or how to evaluate the strategy.
The generation process was done thanks to Apache POI API. It allows to export any data to the documents of Microsoft Word or Excel formats. In this case, there was used the XWPF class, which is responsible for **.docx** file types operations. The Figure 19 presents, how it was implemented.

```java
@Override
public void generateDocument(Long strategyId){
    Strategy strategy = strategyRepository.findById(strategyId).get();
    String output = strategy.getName() + ".docx";
    XWPFDocument document = new XWPFDocument();
    XWPFParagraph title = document.createParagraph();
    title.setAlignment(ParagraphAlignment.CENTER);
    XWPFRun titleRun = title.createRun();
    titleRun.setText(strategy.getName());
    titleRun.setColor("000000");
    titleRun.setBold(true);
    titleRun.setFontFamily("Arial");
    titleRun.setFontSize(20);

    XWPFParagraph missionTitle = document.createParagraph();
    missionTitle.setAlignment(ParagraphAlignment.CENTER);
    XWPFRun missionTitleRun = missionTitle.createRun();
    missionTitleRun.setText("Mission");
    missionTitleRun.setColor("000000");
    missionTitleRun.setFontFamily("Arial");
    missionTitleRun.setFontSize(20);

    XWPFParagraph mission = document.createParagraph();
    mission.setAlignment(ParagraphAlignment.CENTER);
    XWPFRun missionRun = mission.createRun();
    missionRun.setText(strategy.getMission());
    missionRun.setColor("000000");
    missionRun.setFontFamily("Arial");
    missionRun.setFontSize(12);
```

Figure 19: *generateDocument()* method code fragment [Source: Own work]

As one can see, firstly the *Strategy* object is retrieved. Then, the data obtained from previous steps of strategy generation is put into the document. The next parts of the code are repetitive. For each part of generated text, there have to be defined both *XWPFParagraph* and *XWPFRun* object. They are responsible for the text formatting in the resulting document. It is set thanks to the methods such as *setText(), setColor(), setBold(), setFontFamily()* and *setFontSize()*. The method contains a lot of similar code since there are many sections in the document, but there is no other way to reduce its amount.
In the same method there is another interesting code fragment, which is responsible for displaying the long term objectives sorted by the domain. It is presented in the Figure 20.

```
361
362         List<LongTermObjective> objectives = longTermObjectiveRepository.findAllByStrategyId(strategyId);
363         Map<String, List<LongTermObjective>> groupedAndSortedObjectives = objectives
364                 .stream()
365                 .filter(o -> o.getDomain() != null)
366                 .sorted(Comparator.comparing(LongTermObjective::getDescription))
367                 .collect(groupingBy(o -> o.getDomain().getName()));
368
369         for (Map.Entry<String, List<LongTermObjective>> entry : groupedAndSortedObjectives.entrySet()) {
370             String domainName = entry.getKey();
371             List<LongTermObjective> objectivesByDomain = entry.getValue();
372             prepareParagraph(document, domainName, fontSize: 14, pageBreak: false);
373
374             for (LongTermObjective objective : objectivesByDomain) {
375                 prepareParagraph(document, objective.getDescription(), fontSize: 11, pageBreak: false);
376             }
377         }
```

Figure 20: Code fragment responsible for sorting the objectives [Source: Own work]

At the beginning there is created the *objectives* variable of *List* type, which stores all the objectives for the specific strategy. Right after, the list is manipulated thanks to Java 8 stream API. It allows to go through all objects of the list quickly and perform specific operations. In this case, first the objects that have no domain are filtered out and then they are sorted alphabetically by their descriptions. At the end, they are gathered and grouped by each domain name, thanks to the *groupingBy()* method.
Since the key values are now the domain names, one can easily display the objectives sorted by domains. It is done thanks to the *for* loop, which iterates through each key-value pair and firstly generates a section title in the document thanks to *prepareParagraph()* method. Later on there is another *for* loop, which for all of the objectives stored in the map entry writes them down into the document using the previously mentioned method.
In the Figure 21 there is presented the page with sorted objectives of the **.docx** document, which was generated using the *generateDocument()* method.

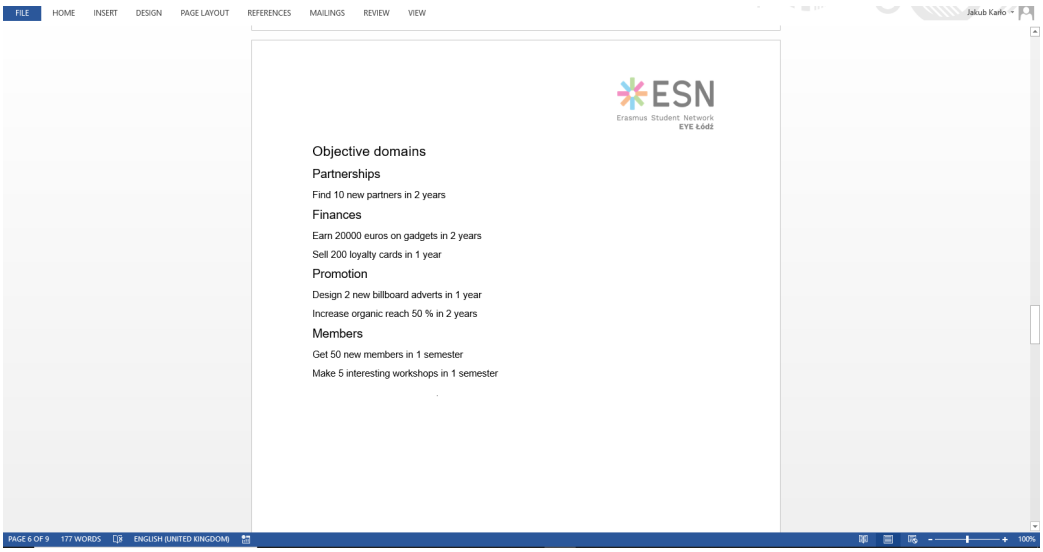


Figure 21: File generated using Apache POI [Source: Own work]

The outcome of generation gave great results. The objectives are properly sorted plus document looks neat and well organised. It is certain that the next teams, which will develop the strategy for ESN-EYE, will find it helpful.

4.2 List management

List management is one of the most important features of the application. The same way of operation is implemented across PEST factors, SWOT factors and also Long Term Objectives. Since Thymeleaf does not allow to pass the list directly to the controller, there has to be implemented a wrapper in order to manage the data input by user properly. In the Figure 22 there is shown the code of one of the wrappers.

```java
package com.eye.esn.domain.wrapper;

import com.eye.esn.domain.PestFactor;

import java.util.List;

public class PestFactorsWrapper {

    private List<PestFactor> factors;

    public List<PestFactor> getFactors() { return factors; }

    public void setFactors(List<PestFactor> factors) { this.factors = factors; }

    public PestFactorsWrapper(List<PestFactor> factors) { this.factors = factors; }

    public void addFactor(PestFactor factor) { this.factors.add(factor); }
}

```

Figure 22: *PestFactorsWrapper* class code [Source: Own work]

As one can see, the class is just storing the list instance. It is initialized in the constructor plus contains just getter, setter and method responsible for adding a new factor. Consequently, it was possible to contain the reference to it in the Thymeleaf code as shown in the Figure 23.

```
<div class="col-md-12">
    <form th:object="${factors}" th:action="@{/strategy/{strategyId}/pest/political(strategyId=${strategy.id})}"
          method="post" class="form">
        <button class="form-add-item-button" type="submit" name="addRow">Add factor</button>
        <div th:each="factor, itemStat : *{factors}">
            <input type="text" th:field="*{factors[__${itemStat.index}__].description}">
            <input type="hidden" class="input-hide" th:field="*{factors[__${itemStat.index}__].id}">
            <input type="hidden" class="input-hide" th:field="*{factors[__${itemStat.index}__].type}">
            <input type="hidden" class="input-hide" th:field="*{factors[__${itemStat.index}__].pestId}">
            <button class="form-remove-button" type="submit" name="removeRow" th:value="${itemStat.index}">x</button>
        </div>
        </br>
        <button class="form-submit-button" type="submit" name="save">Submit factors</button>
    </form>
</div>
```

Figure 23: *pest_factors_political.html* file code [Source: Own work]

In order to access the list, there has to be added an *itemStat* iterator, which goes through *factors* list and its properties. If one wanted just to

display the list elements, there would be no necessity of the wrapper. However, to bind the whole list to the controller, the wrapper is a must, since the list cannot be passed by itself, it has to be held by an object [34]. Then one is able to edit the properties of the specific list item.
What is more, the url called by the *th:action* property can call three different methods from the controller. The key is the *name* parameter for each button. It keeps the Thymeleaf code cleaner, as there is no need of having three different urls and controller can easily recognize, which method should be called thanks to the *params* annotation in the Java code, as shown in the Figure 24.

```
93      @PostMapping(value = "/{id}/pest/{factorType}", params = {"save"})
94  @   public String savePestFactorsForType(@PathVariable Long id, @PathVariable String factorType, @ModelAttribute PestFactorsWrapper factors) {
95          strategyService.savePestFactors(factors.getFactors(), id, factorType);
96          return "redirect:/strategy/{id}/pest";
97      }
98
99
100     @RequestMapping(value = "/{id}/pest/{factorType}", params = {"addRow"})
101 @   public String addPestFactor(@PathVariable Long id, @PathVariable String factorType, final PestFactorsWrapper pestFactorsWrapper,
102                               final BindingResult bindingResult, Model model) {
103         model.addAttribute( attributeName: "strategy", strategyService.getStrategyDetails(id));
104         strategyService.preparePestFactors(pestFactorsWrapper);
105         model.addAttribute( attributeName: "factors", pestFactorsWrapper);
106         return "pest_factors_" + factorType + "";
107     }
108
109     @RequestMapping(value = "/{id}/pest/{factorType}", params = {"removeRow"})
110 @   public String removePestFactor(@PathVariable Long id, @PathVariable String factorType,
111                                   final PestFactorsWrapper pestFactorsWrapper, final BindingResult bindingResult, Model model,
112                                   final HttpServletRequest req) {
113         final Integer rowId = Integer.valueOf(req.getParameter( name: "removeRow"));
114         strategyService.removePestFactor(pestFactorsWrapper.getFactors().get(rowId.intValue()).getId());
115         pestFactorsWrapper.getFactors().remove(rowId.intValue());
116         model.addAttribute( attributeName: "strategy", strategyService.getStrategyDetails(id));
117         model.addAttribute( attributeName: "factors", pestFactorsWrapper);
118         return "pest_factors_" + factorType + "";
119     }
120
```

Figure 24: *StrategyController* code [Source: Own work]

Basing on that annotation, the method responsible for either saving, adding or removing the list items is called. Digging deeper into each of them, saving is just passing the whole list of entities and saving it in the database. The other two operations are a bit more complicated. While removing the list item, one has to remember that it should be also deleted from the database before the list is saved, since the repository *save()* method would not remove the ones deleted from the view. That is why, before removing the item from the wrapper object, there has to be called *removePestFactor()* method. Then it is certain that all unnecessary factors would not remain in the database.

Then the page is reloaded and the same Thymeleaf template is rendered. When it comes to adding the new item, it is added to the wrapper object list without saving it immediately, as saving the whole list at once is more efficient. The only constraint in this case is to check, if the list of items actually exist and, if not, initialize it, what is done in the *preparePestFactors()* method. Its code is presented in the Figure 25.

```
128     @Override
129     public void preparePestFactors(PestFactorsWrapper pestFactorsWrapper) {
130         if (pestFactorsWrapper.getFactors() == null) {
131             ArrayList<PestFactor> emptyFactorsList = new ArrayList<>();
132             pestFactorsWrapper.setFactors(emptyFactorsList);
133         }
134         PestFactor emptyFactor = new PestFactor();
135         emptyFactor.setDescription("");
136         pestFactorsWrapper.getFactors().add(emptyFactor);
137     }
```

Figure 25: *preparePestFactors()* method code [Source: Own work]

The method performs simple null check on the list in the wrapper object and creates a new one if necessary. Right after there is added a new *PestFactor* with an empty description. Then the page is re-rendered and user is able to write a factor description.
As mentioned before, SWOT factors base exactly on the same operations, as well as long term objectives. Therefore, the list management using wrappers is one of the core elements of the application.

5 User Guide

In the following paragraphs there are described all functionalities of the developed application in order to inform the potential user how to work with it.

5.1 Login

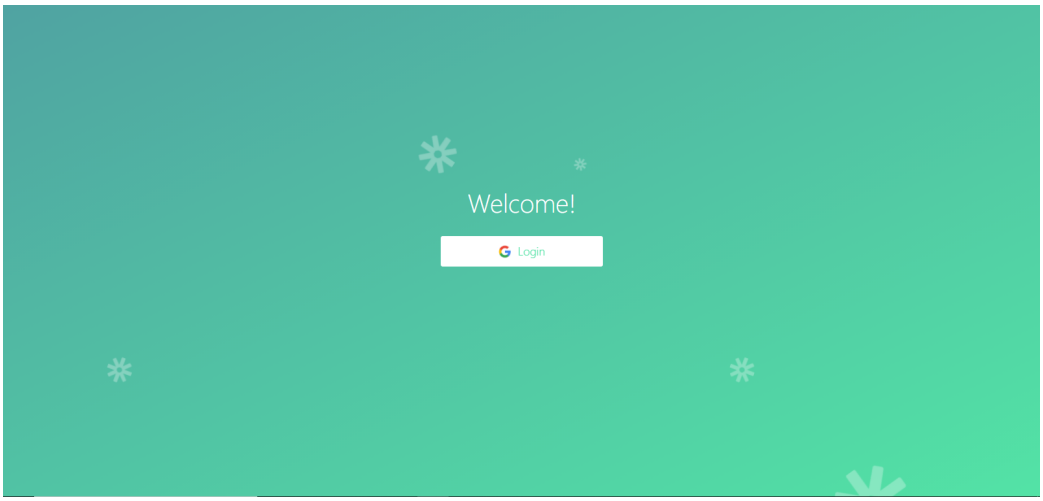


Figure 26: Login page [Source: Own work]

When user tries to access any url of the application, he needs to be authenticated. In order to do so, he is redirected to the login page. It contains a button, which is responsible for entering the Google Oauth2 service, and an animated stars going up the page. The stars are the ones from ESN logo. When the button is pressed, the use is redirected to the authentication provider page and after logging in with Google, he is redirected to the welcome page with the user's name, email and profile picture from Google, which is presented in the Figure 27. The page is displayed for two seconds and then the strategy list is shown.

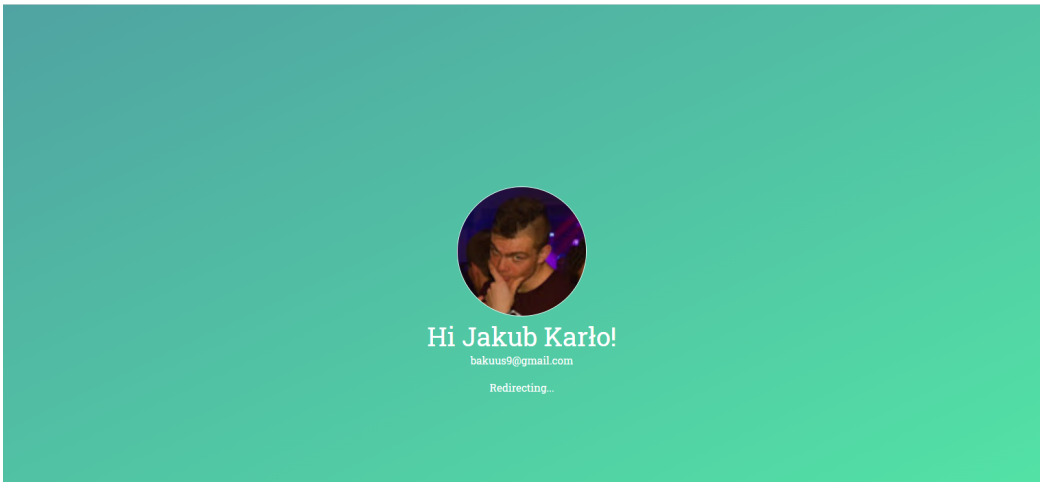


Figure 27: Welcome page [Source: Own work]

5.2 Strategy list

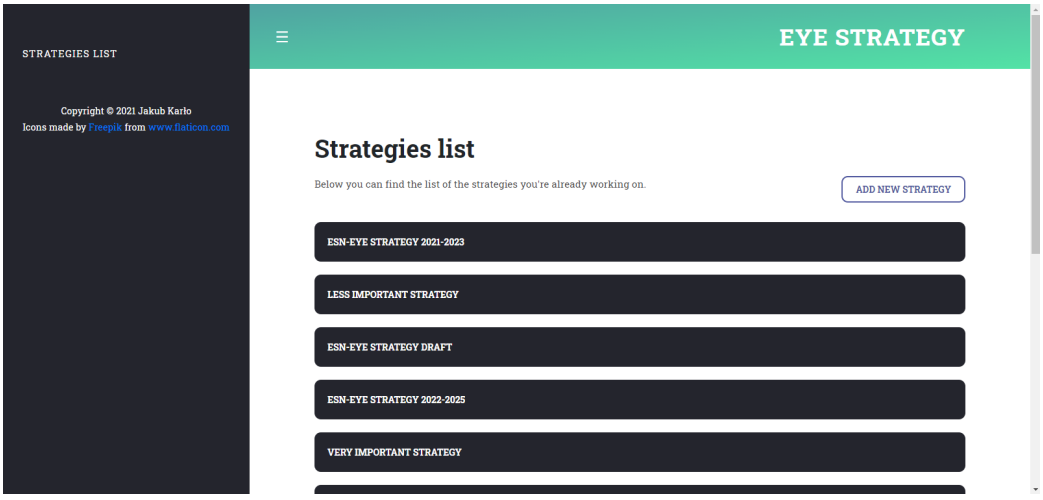


Figure 28: Strategy list page [Source: Own work]

In the strategy list page there is displayed a list of all strategies user has created. They are sorted from the newest to the oldest. The list is displayed

in form of buttons, which redirect to the specific strategy dashboard page. On the top right corner there is a button which allows to add the new strategy. When it is pressed, there is displayed the page responsible for addition of new strategy, as shown in the Figure 29.

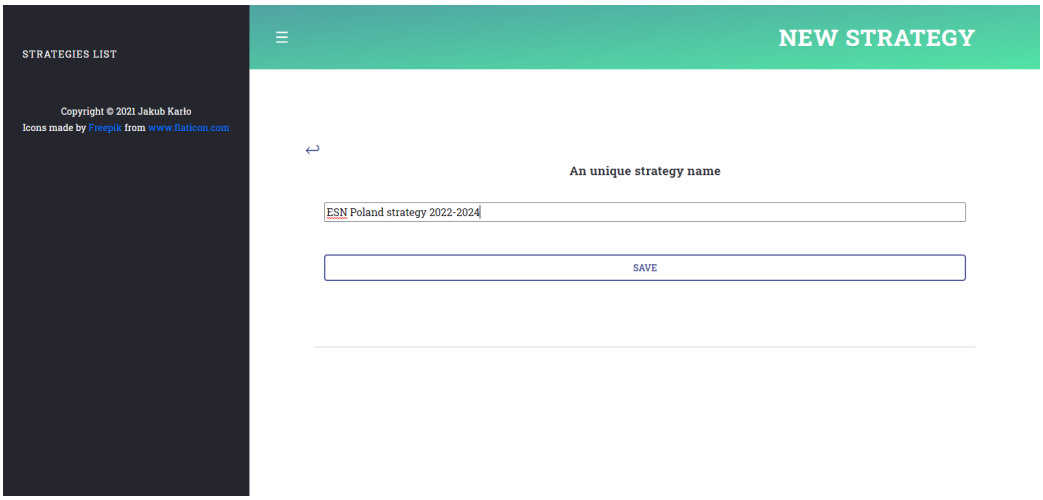


Figure 29: New strategy page [Source: Own work]

As it is clearly visible, the only data that has to be provided is the strategy name. When a new strategy is added it is displayed on the top of the list, as presented in the Figure 30.

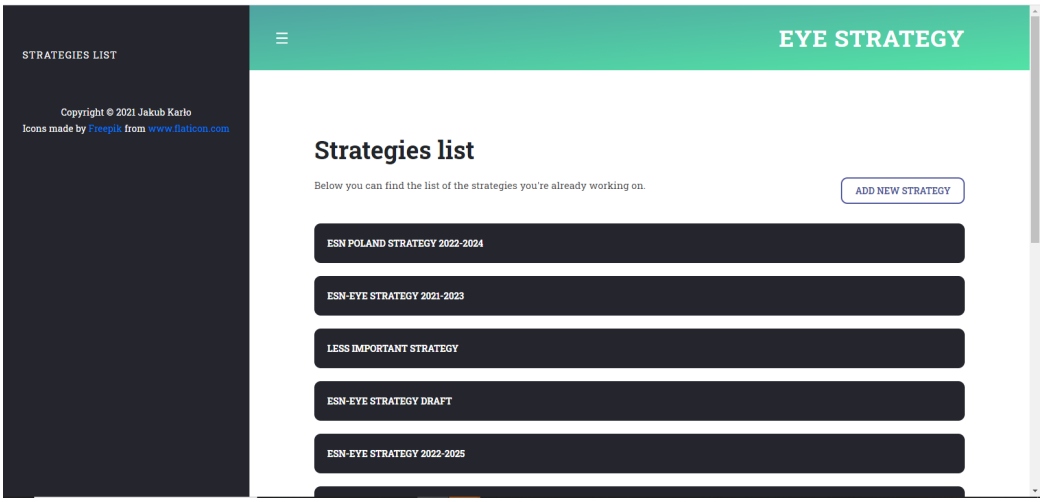


Figure 30: Strategy list page with new strategy added [Source: Own work]

What is more, on the left side of the screen there is displayed a sidebar, which can be also hidden thanks to the hamburger button. This component is present in all pages of the application. It may seem empty, since in the strategy list page there is only the link to the page itself. However, its content varies basing on the page that the user is currently accessing. In general, the sidebar was designed to display the information that could be helpful for the user during strategy development steps. It would be shown in the next paragraphs.

5.3 Strategy dashboard

As it was previously mentioned, when the user clicks on any of the buttons with strategy names in the strategy list, there is performed a redirection to the strategy dashboard. The screenshot of one of them is presented in the Figure 31.

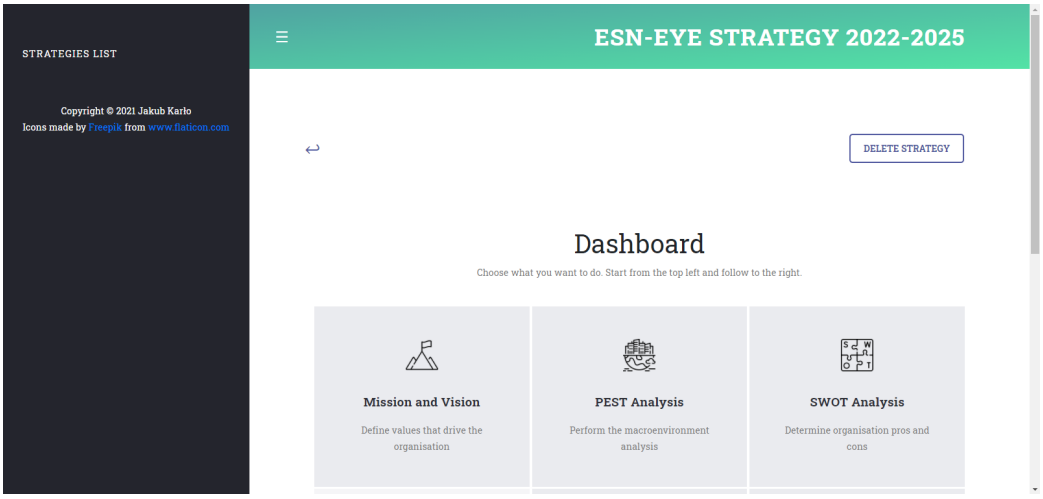


Figure 31: Strategy dashboard page [Source: Own work]

As one can see, on the top bar there is displayed the name of the strategy and right below there are two buttons. The arrow on the left allows to come back to the strategy list and the right one lets user delete the current strategy. If the user decides to press this button, there is shown a modal dialog, presented in the Figure 32, which requires the confirmation in order not to accidentaly delete something that user worked on for hours.

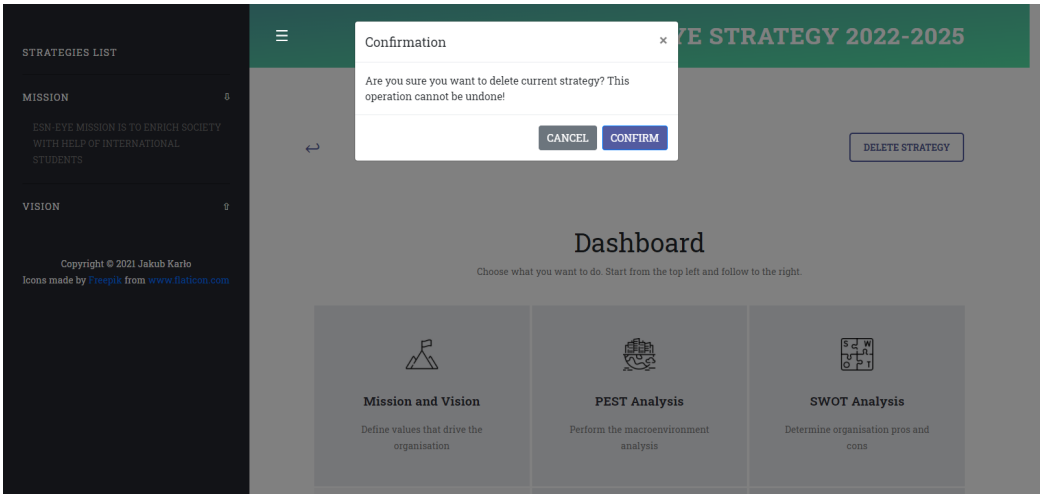


Figure 32: Strategy removal confirmation modal [Source: Own work]

In the Figure 31 there was also present a dashboard title with couple of buttons in forms of tiles. They allow user to perform the specific steps of strategy development. All of them are shown in the Figure 33.

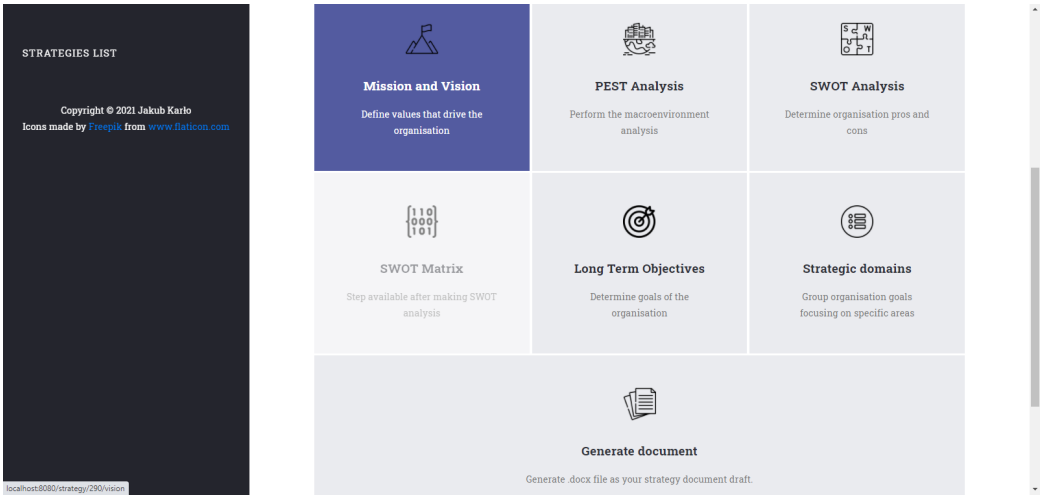


Figure 33: Strategy dashboard tile buttons [Source: Own work]

There are seven tiles in total. The user can access and edit any of the

strategy development stages anytime, even after performing the next step. However, there are two exceptions. They are both connected to SWOT analysis. Firstly, until the SWOT analysis is at least created, the SWOT matrix tile is disabled, hence it will miss the necessary data. Secondly, after creating the SWOT matrix, there is no option to go back to SWOT analysis. In order to perform the strategy creation process properly, the steps should be conducted from the left top corner to the right. In the following paragraphs there are described all of them.

5.4 Mission and Vision

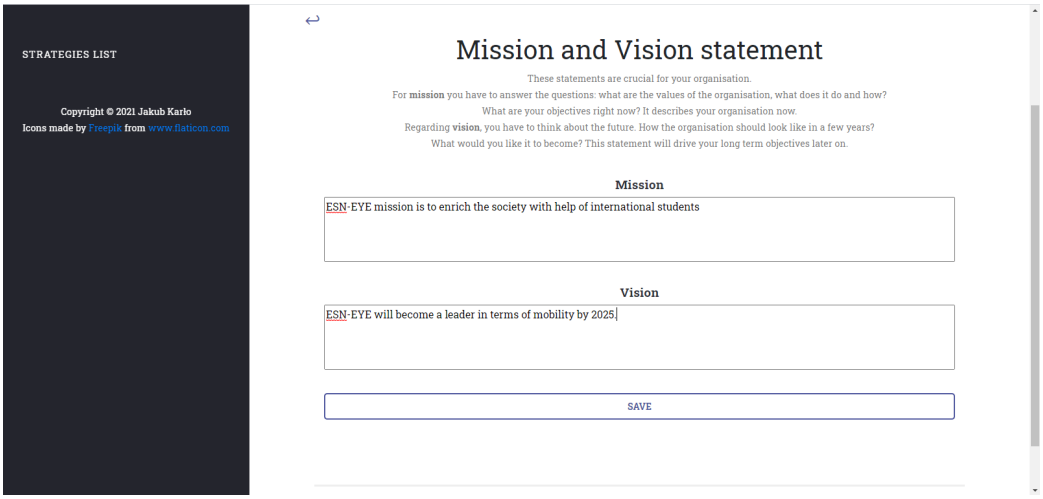


Figure 34: Mission and vision page [Source: Own work]

All of the analysis pages have a similar layout to the one presented in 34. They contain the back button, which allows to return to the strategy dashboard, the analysis title and helpful description below. In case of mission and vission, there are written some questions and information, which will lead user to the formulation of these two important statements.
After the statements are formulated and saved, the user can see them on any other page on the sidebar, what is presented in the Figure 35.

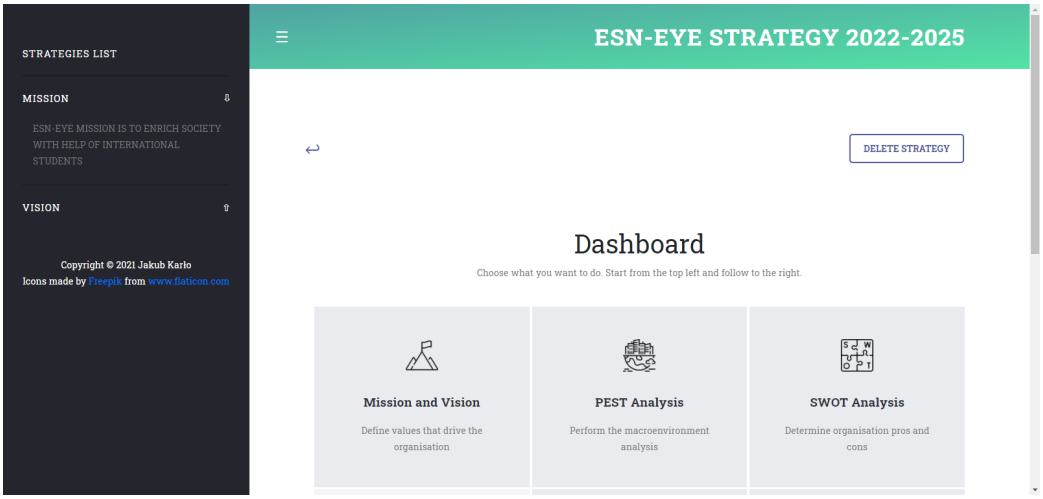


Figure 35: Strategy dashboard with sidebar hints [Source: Own work]

5.5 PEST analysis

After choosing the PEST analysis option, one can see the following screen.

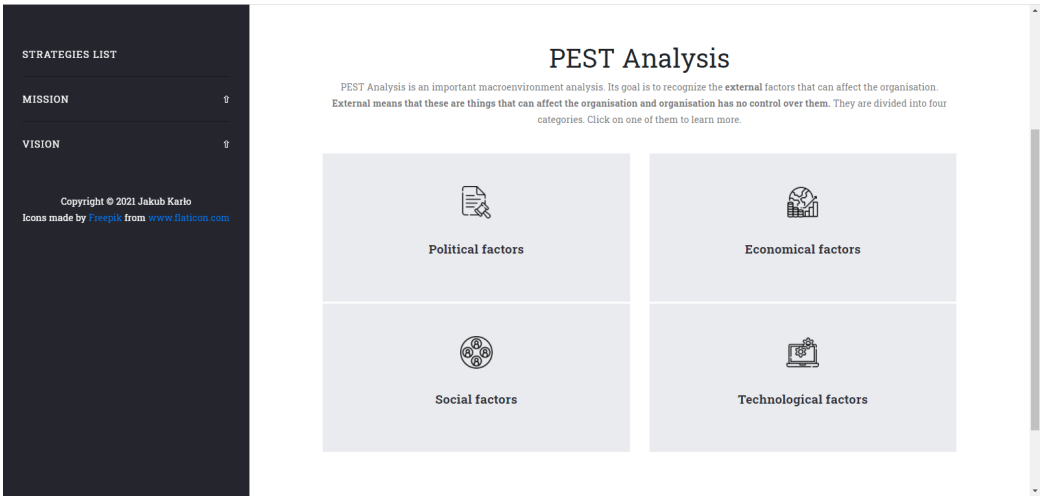


Figure 36: PEST analysis dashboard [Source: Own work]

The PEST analysis dashboard consists of four interactive tiles. They

allow the user to edit the list of the political, economical, social and technological factors. After clicking on any of the options, there is displayed a page shown in the Figure 37.

Figure 37: PEST factors page [Source: Own work]

An example shown in the Figure 37 is the political factors page, however, for all other types the layout is the same, only the hint differs. The user is able to add and remove factors dynamically and when they are saved, he is redirected back to the dashboard and all factors are displayed on the top of the page to let an user have an overview. It is presented in the Figure 38.

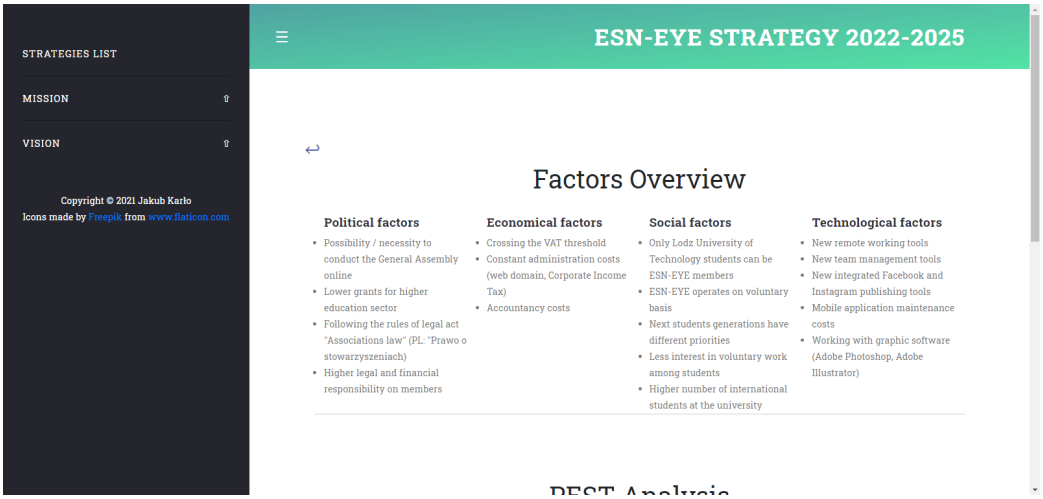


Figure 38: PEST factors overview [Source: Own work]

5.6 SWOT analysis

SWOT analysis steps are performed in the same way as the PEST analysis ones. The difference is that this time the factors are internal. It means that user this time should figure out the strengths, weaknesses, opportunities and threats. The dashboard is presented in the Figure 39.

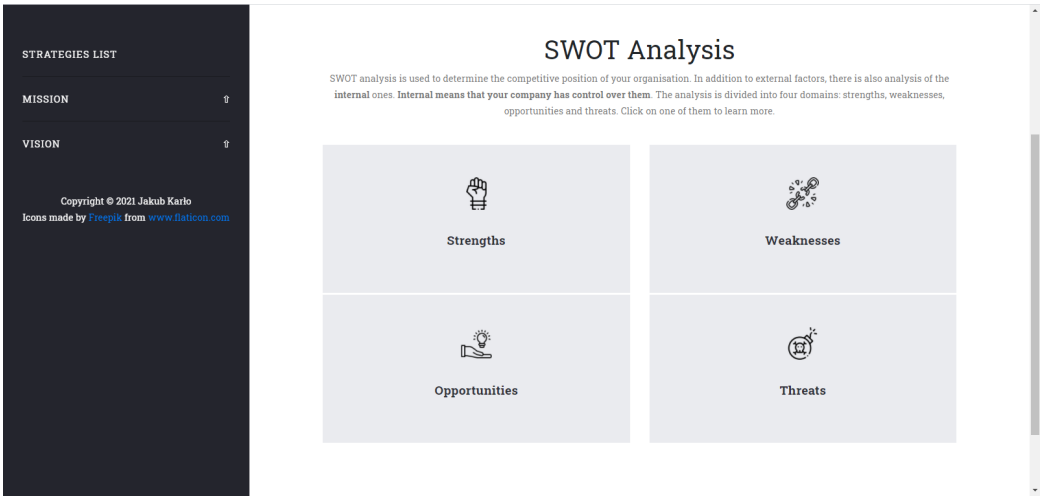


Figure 39: SWOT analysis dashboard [Source: Own work]

After clicking on any of the available options the user is redirected to the page with layout shown in the Figure 40.

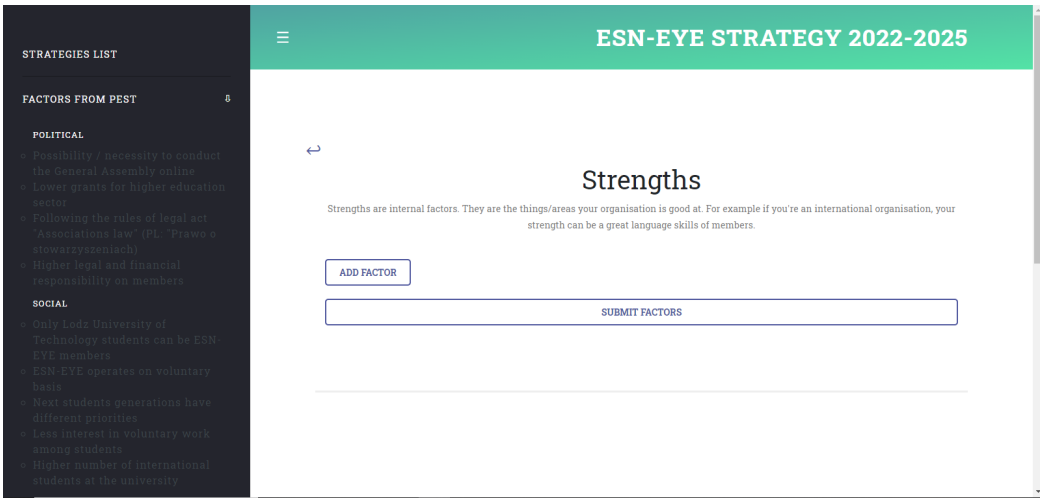


Figure 40: SWOT factors page [Source: Own work]

As it was with any of the PEST factors group, the user can also dynamically add and remove them. The important feature is that all the results from the

PEST analysis are displayed on the sidebar in order to remind the results from the previous step. Similarly to the analysis conducted before, after submitting the list, the factors are displayed on the top of the dashboard, as presented in Figure 41.

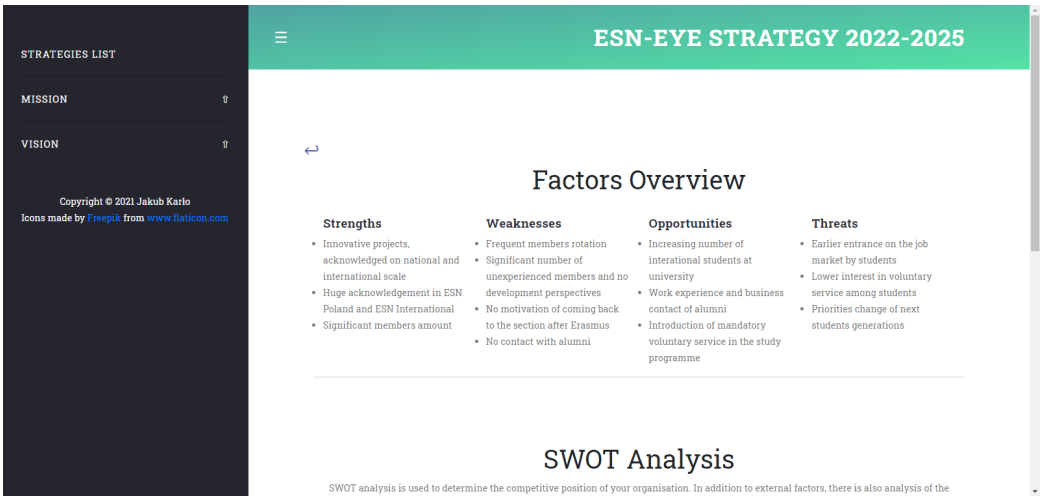


Figure 41: SWOT factors overview [Source: Own work]

5.7 SWOT matrix

After conducting the SWOT analysis, the next step is to create a SWOT matrix. It is extremely important to be sure that the SWOT analysis is complete, since after creating the matrix, there is no chance of editing the previous analysis. In order to prevent the user from accidental choice, there is displayed a confirmation dialog, as shown in the Figure 42.

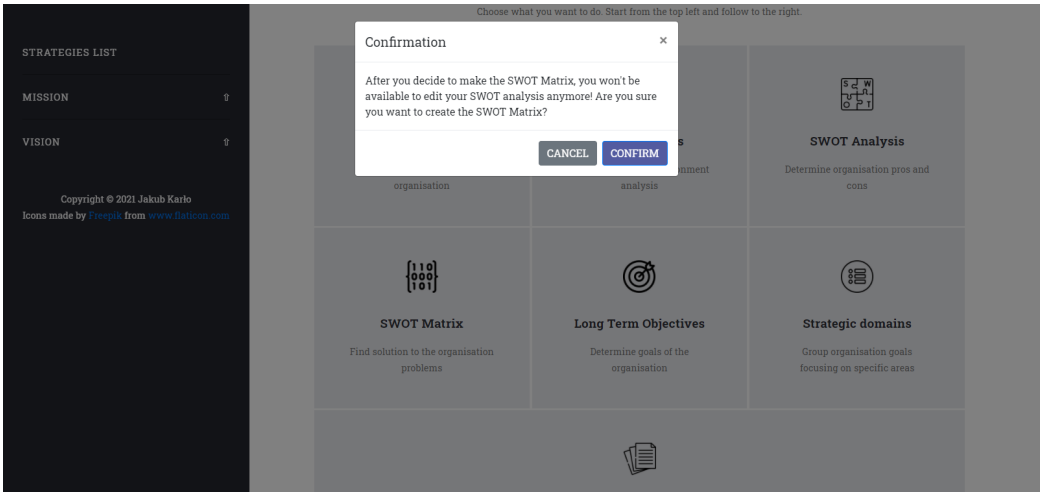


Figure 42: SWOT matrix creation confirmation dialog [Source: Own work]

When user click "confirm", there is a new analysis object created in the background and one can see a dashboard shown in the Figure 43.

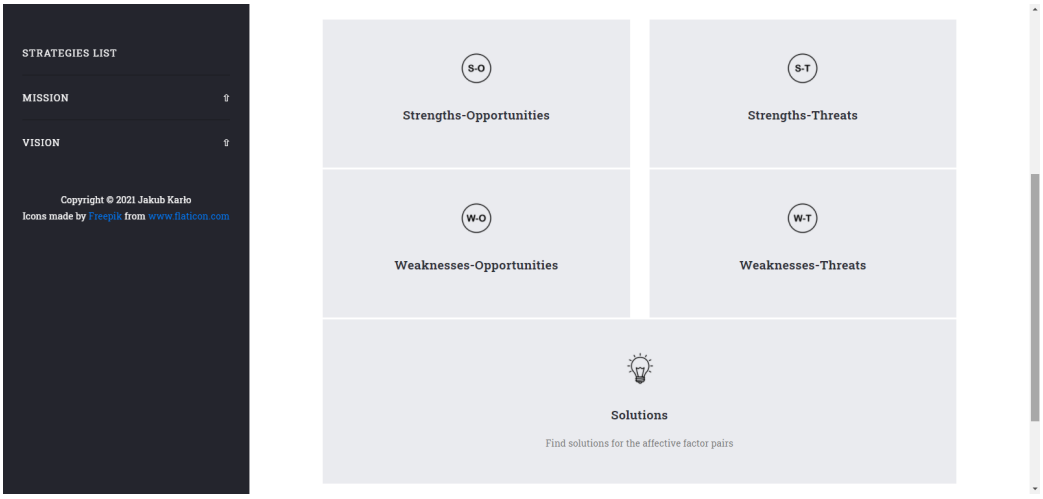


Figure 43: SWOT matrix dashboard [Source: Own work]

As one can see, the dashboard is similar to the previous analysis. Once again there are four basic steps to perform, which include analysis of weaknesses

and strengths set against opportunities and threats. In addition, there is also the tile responsible for finding the solutions for the previously defined problems, what will be described later. When user chooses one of the other available options, there is shown a page presented in Figure 44.

Figure 44: SWOT matrix affection levels page [Source: Own work]

In that page, the user should choose if each of two factor pairs influence each other. There are three possible options: HIGH, LOW and NONE. Therefore, if strength and opportunity have a great impact on each other, there should be chosen HIGH level. If they influence each other, but it is not that significant, one should choose LOW option. Finally, if one cannot determine any influence between them, the last option, which is NONE, should be chosen.
After choosing levels for all of the factor pairs, user should save the progress using the corresponding button. After returning to the dashboard and determining all affection levels between factors, one should follow it up with the solutions.
In order to do so, the tile with name "solutions" should be clicked. Consequently, there is displayed a page shown in the Figure 45.

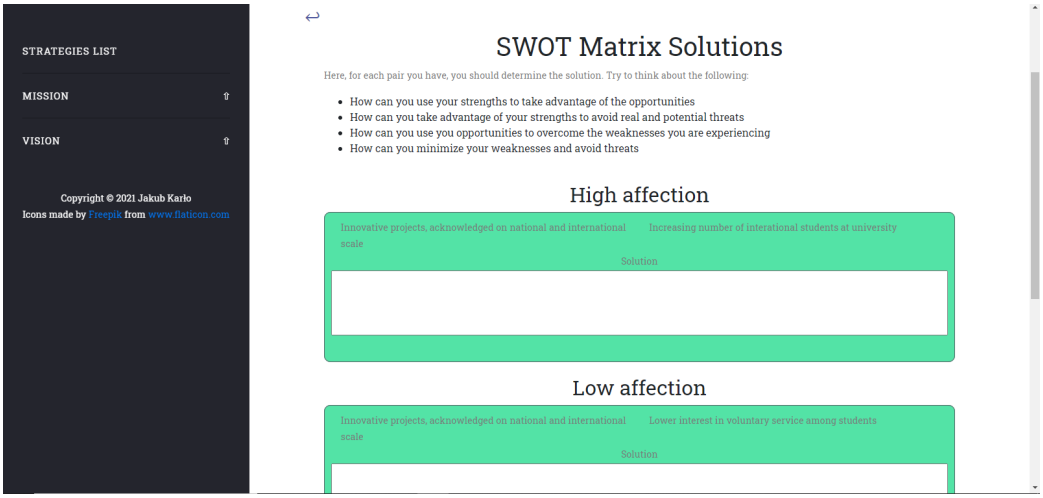


Figure 45: SWOT matrix solutions page [Source: Own work]

In this page, there are shown the factor pairs, for which there is either low or high affection between them. At the top, there is shown the hint, what kind of solution should be found for any of the corelation types, and it is as follows:

- how one can use strengths to take advantage of the opportunities,
- how one can take advantage of strengths to avoid real and potential threats,
- how one can use opportunities to overcome the weaknesses being experienced,
- how one can minimize weaknesses and avoid threats.

All of the answers for the following questions should be noted down in the "solution" textareas in corresponding list items. Finally, as in the previous steps, one should press "save" button in order to finish the analysis.

5.8 Long term objectives

After saving the solutions from SWOT matrix, one can proceed with formulation of long term objectives. In order to do so, one should choose the right option

from the strategy dashboard. It will result in the page shown in the Figure 46.

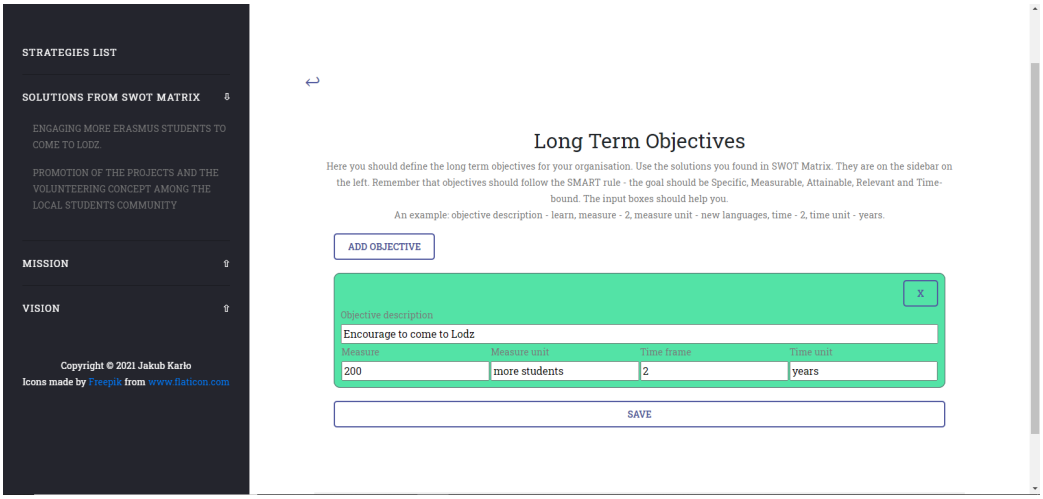


Figure 46: Long term objectives page [Source: Own work]

As one can notice, similarly to PEST and SWOT analysis, user can add and remove the objectives as he wants. The description under the page title states that each goal should follow the SMART principle. It was explained in detail in the Section 2.1.5. Each objective consists of the description, measure, measuring unit, time and time unit. Measure and time are the numbers, while the measuring and time units can be anything that user wants. It is easier to figure out how to do it properly with an example. Imagine that one would like to release 20 new songs in 2 years. Then in the boxes of the long term objective form should be put the following values:

- description - release,
- measure - 20,
- measure unit - new songs,
- time - 2,
- time unit - years.

What should be really helpful, is that on the sidebar there are displayed the solutions from the SWOT matrix analysis. The goals should be formulated basing on them. Then as always, the progress should be saved using the proper button at the bottom.

5.9 Objective domains

The last analysis step is to group the goals by the strategic domain. In order to do so, one should choose option "strategic domains" from the strategy dashboard and the resulting page will look like in the Figure 47.

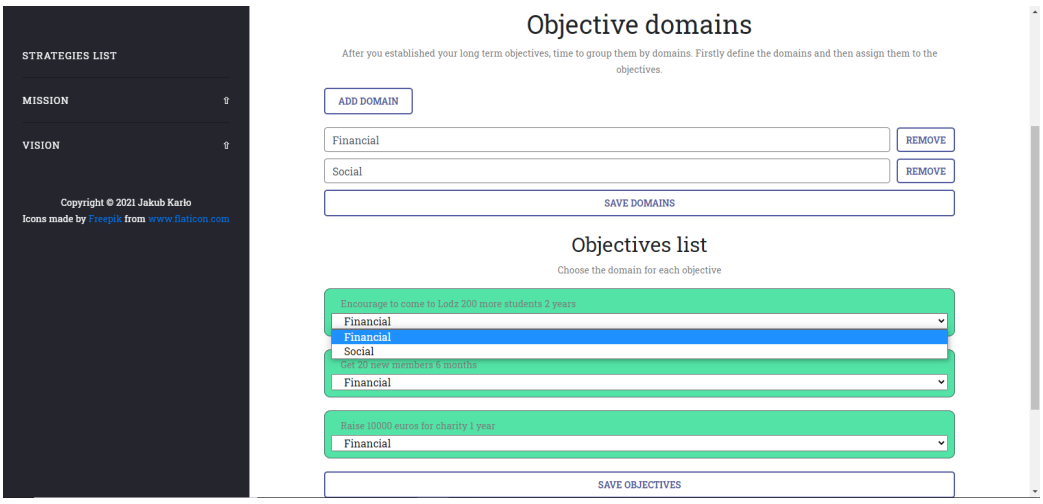


Figure 47: Objective domains page [Source: Own work]

In that page there are displayed two forms. First is responsible for adding the domains, while the second one displays all the previously formulated objectives and allows user to choose from of the domains addded in the form above. Therefore, one should look at the goals closely and find something that they have in common. For example in the Figure 47, two first objectives connect to the members and students. Basing on that, one can determine, that the domain for them should be "social", so he adds it to the domain list and later is able to choose it for the previously mentioned objectives. When everything is completed, one should save the results and whole analysis process is finished. Then there the last step is to generate the document itself.

5.10 Generate document

When user chooses the last tile, there will be displayed a saving prompt, which will allow to save the document in the operating system. The default filename is the name of the strategy. It is presented in the Figure 48.

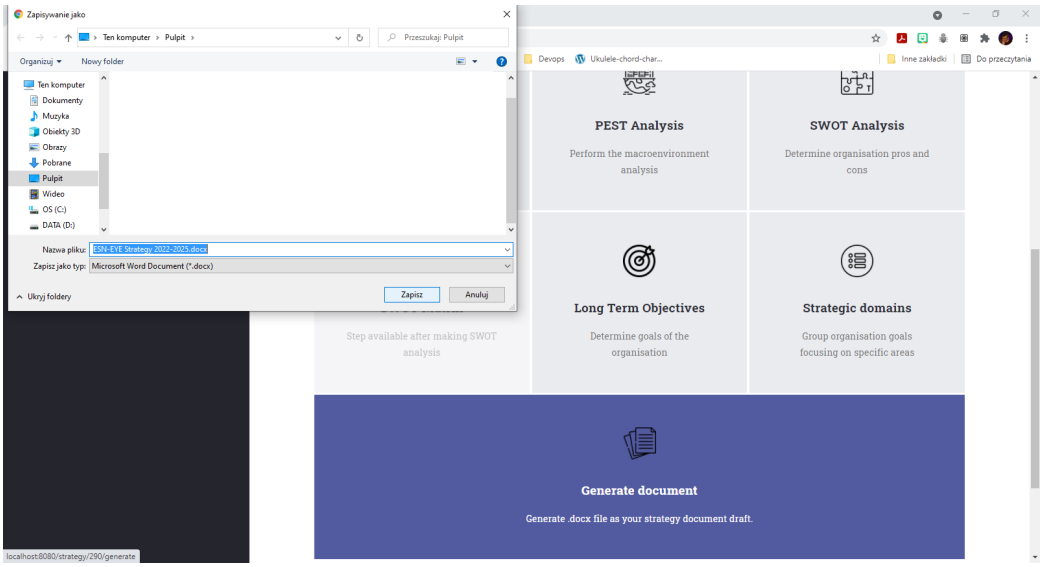


Figure 48: Saving document prompt [Source: Own work]

This step is the last one, what means that the strategy generation is complete. Since the document is a template, there still has to be some effort put, especially in terms of the descriptive parts of the document, such as current organisation situation or the evaluation guidelines. The generated document template is included in the appendix section.

6 Strategy development steps

In this section there are presented results of all analysis performed with the help of the developed application, as well as necessary consultations. Some of the analysis results had to be translated from Polish, since ESN-EYE members are Polish citizens and consultation session conducted with them was done using their mother tongue.

6.1 Mission and Vision

6.1.1 ESN-EYE Mission

Since ESN-EYE is a member of Erasmus Student Network community, the mission statement is the same as the whole network:
Enrichment of society through international students [35].

6.1.2 ESN Vision

By 2025, ESN will be the global network of the Erasmus Generation, committed to improving international education and providing self-development opportunities to two million young people, fostering intercultural understanding and creating positive change in society [35].

6.1.3 ESN-EYE Vision

ESN-EYE has to follow the vision of the whole network, however it can be more specific based on section local environment. Therefore, the vision statement sounds as follows:
By 2025, ESN-EYE will be a leader in the field of the students mobility among the Lodz University of Technology student organisations, engaged into international education promotion, ensuring the possibilities of self-development and intercultural understanding, as well as being an initiator of introducing the positive changes leading to society internationalization.

6.2 Consultations with members

In order to get the bigger picture of how organisation should look like in 3 years time, there was conducted a consultation session with ESN-EYE members. It was done thanks to the Socrative tool [36], which allows to

gather the answers for the questions and generate the report.
All questions were concentrating on the next three years. They were as follows:

- What kind of activity will be ESN-EYE conducting?
- What will be the section structure?
- What kind of people will become ESN members?
- How will the section financial situation look like?
- Which remote work tools/communication channels will section use?
- How will the section promotion ways look like?
- How will ESN-EYE be perceived by the university staff and students?
- What kind of partners the section will cooperate with?
- What should be done to maintain the friendly atmosphere in the section?

The generated report with results is attached in the appendix section. The results became useful especially during the SWOT analysis and objectives formulation.

6.3 PEST Analysis

In the next sections there are presented the outcomes of PEST analysis based on the own knowledge and one gathered during the consultation sessions.

6.3.1 Political factors

- Possibility / necessity to conduct the General Assembly online
- Lower grants for higher education sector
- Following the rules of legal act "Associations law" (PL: "Prawo o stowarzyszeniach)
- Higher legal and financial responsibility on members
- Introducing the obligatory voluntary duties in study programmes

- Auction procedures
- Cooperation contract with International Cooperation Centre (ICE)
- Data sharing contract with ICE
- RODO regulations
- Higher number of responsibilities due to cooperation with university, alongside with ESN Poland and ESN International membership.
- Worse relations between European Union and Poland

6.3.2 Economical factors

- Crossing the VAT threshold
- Constant administration costs (web domain, Corporate Income Tax)
- Accountancy costs

6.3.3 Social factors

- Only Lodz University of Technology students can be ESN-EYE members
- ESN-EYE operates on voluntary basis
- Next students generations have different priorities
- Less interest in voluntary work among students
- Higher number of international students at the university
- Growing demand for students on the job market
- Job market is more attractive for students each year
- Members consider themselves less and less as an organisation member each year
- Losing the idea of students organisation, becoming corporation-like
- Stronger position of the organisation at the Lodz University of Technology

- Huge ordinary and board members rotation
- Companies focus on soft skills more
- No possibility of conducting events (epidemic state, war etc.)
- In 2022/2023 academic year there will come doubled amount of students
- "Eco" trend in the world
- Students are trending to use different social media than Facebook, mostly Instragram and TikTok
- Nationalistic/Xenophobic trend in Poland

6.3.4 Technological factors

- New remote working tools
- New team management tools
- New integrated Facebook and Instagram publishing tools
- Mobile application maintenance costs
- Working with graphic software (Adobe Photoshop, Adobe Illustrator)
- Online events

The analysis results focused mostly on political and social factors, since ESN is an organisation, which operates at high education area and focuses on society enrichment, as stated in the mission. A lot of factors are associated with current and future members, what in terms of high rotation is the key.

6.4 SWOT Analysis

In the next sections there are presented the outcomes of SWOT analysis.

6.4.1 Strengths

- Innovative projects, acknowledged on national and international scale
- Huge acknowledgement in ESN Poland and ESN International
- Great relations with the university
- Significant members amount
- Friendly atmosphere in the section
- Clear organisation structure
- Stable financial situation

6.4.2 Weaknesses

- Frequent members rotation
- Significant number of unexperienced members and no development perspectives
- No motivation of coming back to the section after Erasmus
- No contact with alumni
- Mess in the members database and documents
- No strategic partners
- No long term planning, for example in terms of promotion
- No motivation among members
- Insufficient knowledge transfer between members
- Low recognition among Lodz University of Technology students

6.4.3 Opportunities

- Increasing number of interational students at university
- Work experience and business contact of alumni
- Introduction of mandatory voluntary service in the study programme
- Higher demand among companies for still studying employees
- Bigger range of trainings and development courses
- Growing organisation acknowledgement at Lodz University of Technology

6.4.4 Threats

- Earlier entrance on the job market by students
- Lower interest in voluntary service among students
- Priorities change of next students generations
- Weakening identity of a single member as an organisation member
- Higher demands for members from ESN Poland and ESN Interational, as well as university
- Higher legal and financial responsibility for members
- Losing the idea of students organisation and heading for becoming a corporation

Taking a look on the analysis results, the main strengths of ESN-EYE are projects made by their members, who enjoy the friendly atmosphere in section. The drawbacks are mostly focusing on short term goals and projects due to high members rotation, which leads to most of the weaknesses mentioned. In terms of external factors, most opportunities focus on the broader possibilites of self-development for members, including the knowledge transfer from alumni and workshops. When it comes to threats, they seem to mostly focus on losing the identity and having higher responsibilities due to the growing position of the organisation.

6.5 SWOT Matrix

In this section, there are presented the solutions made during the SWOT matrix analysis. The chart with the influence level is attached in the appendix section. It was exported into Excel document to have the more clear picture.

Solutions

- Making the innovative projects cyclic in order to attract potential members,
- Enrichment of the offer for partners,
- Using the innovative projects as an advantage while talking with partners and university,
- Moving some events online in order to reach more people, for example with workshops,
- Using some trends for projects development, for example eco trend,
- Building the image of strong, well organised section, which has a stable position in ESN Poland and ESN International,
- Building the European awareness and feeling of being a European citizen among students,
- Bulding the community of ESN-EYE alumni,
- Making more ESN section cooperations in order to organise workshops together,
- Tightening the cooperation between ESN-EYE and Students Association,
- Cooperating with the university partners,
- Development of the training base for members (webinars etc.),
- Introducing the new technology tools into the section on daily basis, for example during events organisation,
- Introducing new remote work tools,

- Development of the projects and activities, which will focus on personal attitude to every member,
- Engaging the members to help the board members with their daily tasks, for example by making the support groups,
- Paying more attention to integration and building the feeling of identity inside organisation,
- Certifying the skills gained by being at the specific function in organisation,
- Gathering all the knowledge in order to transfer it to new members more efficiently,
- Focusing on the members satisfaction in order to make them stay longer in the section,
- Sending the email newsletter with information about ESN-EYE to students coming back from Erasmus.

There were found a lot of solutions including the strengths enhancement and weaknesses diminishment. It was a crucial step, which allowed proceed with the long term objectives.

6.6 Long-term Objectives

Basing on the solutions made in the SWOT matrix, the long term objectives were formulated and organised into five strategic domains. They are presented in the following sections.

6.6.1 Organisation role among the local society

- Organise 3 big events focusing on eco-friendly life in 2 years.
- Make 2 online workshops with an experienced alumnus each semester.
- Each semester organise a discussion panel about the future of European Union.

6.6.2 Partnerships and collaborations

- Cooperate with the city council to make a one partnership with another European city in 2 years.
- Create a new partnership offer in 6 months and send it every semester to the potential partners.
- Every semester make an innovative project with support of at least one partner.
- Make a section cooperation with ESN PW (Warsaw University of Technology) with hard-skills workshops each semester.
- Find at least one new partner for the university in 2 years.

6.6.3 Members and their development

- Create a mentoring programme for section members and repeat it every semester.
- Make a Facebook post for members about one of the ESN-EYE alumnus at least each month for one year.
- Organise a meeting with alumni at least once a year.
- Create a drive with all the workshops and update it every semester with at least 3 new webinars.
- Each semester every board member should delegate at least 2 tasks to the members of the organisation.
- Organise a meeting for knowledge transfer for all interested members at least twice a semester.
- Find 5 new ways of integrating people in the section in 1 year.
- Increase number of members coming back from Erasmus by 10 in 2 years.

6.6.4 Administration and finance

- Put all the important documents in a digital version on the organisations drive in 2 years.
- Introduce the VAT tax payment procedures in 2 years.
- Allow exchange students to pay for events using card in 2 years.
- Prepare a grant application to get at least 10 000 euros. The application should be done in 2 years.

6.6.5 Promotion

- In 1 year prepare the universal template for certificate confirming the skills gained during being in charge of the section.
- Develop a professional and intuitive website in 2 years.
- Design 6 new ESN-EYE gadgets in 2 years.

7 Summary

The goal of the thesis was to develop and application, which will help the future ESN-EYE Lodz members develop the operational strategy and to prepare the strategy for years 2021-2023 for the organisation. As a result at the beginning there was implemented a Spring Boot application which led the user through the whole process of strategy development. The implementation process required high skills and broad knowledge in Java programming and in Spring Boot framework design principles as well. What is more, the experience in strategic management was also an important part of the development process, since the application needed to contain a lot of hints for the future user. Later on, using the web application there was created a strategy for ESN-EYE Lodz. The help of the developed solution saved a lot of time during the analysis phases.

There was significant amount of resources studied, in order to understand all the available frameworks and find the ultimate solution. Spring Boot, since it was developed, gained a high popularity, what results in lots of practical guides, reference pages and documentations, which helped during the development process. From the point of view of a developer, the theoretical background is as essential as high programming skills. In terms of strategy, it was extremely important to decide which development tools to choose and study them thoroughly to understand their purpose.

The analysis of the existing solutions to the problem was beneficial. Pointing out the advantages and drawbacks of each considered application helped to avoid the mistakes made by the developers of other applications and get inspired by the useful features as well.

The important issue was to thoroughly think the layout of the application to make it intuitive and catchy. All of the components of the interface were designed in order to look coherent. An useful tool was the color wheel, which allowed to find the ideal combination of the colors. As a consequence, the application appears neat and well organised.

Considering the strategy, the analysis performed alongside with the consultations with the members allowed to create the document, that hopefully will be realised properly by board and the members what will result in strengthening organisation position. What is more, the prepared application will be presented to the ESN-EYE board in order to put it on a remote server and allow the next generation of members create the document by themselves.

References

[1] *Difference Between Vision and Mission Statements: 25 Examples*, Available at: https://www.clearvoice.com/blog/difference-between-mission-vision-statement-examples/ (Accessed: 27.04.2021)

[2] *Analiza PEST - Encyklopedia Zarzadzania*, Available at: https://mfiles.pl/pl/index.php/Analiza_PEST (Accessed: 27.04.2021)

[3] *Analiza SWOT - Encyklopedia Zarzadzania*, Available at: https://mfiles.pl/pl/index.php/Analiza_SWOT (Accessed: 27.04.2021)

[4] *How to Set SMART Goals*, Available at: https://www.elegantthemes.com/blog/business/smart-goals (Accessed: 28.04.2021)

[5] *What is the Balanced Scorecard?*, Available at: https://www.intrafocus.com/balanced-scorecard/ (Accessed: 28.04.2021)

[6] *Best Strategic Planning Software 2021 — Reviews of the Most Popular Tools & Systems*, Available at: https://www.capterra.com/strategic-planning-software/#top-20 (Accessed: 26.04.2021).

[7] *monday.com — Work Without Limits*, Available at: https://monday.com/ (Accessed: 26.04.2021).

[8] *Enterprise Performance Management and BI unified in one seamless solution*, Available at: https://www.jedox.com/ (Accessed: 26.04.2021).

[9] *Versatile & Robust Project Management Software — Wrike*, Available at: https://www.wrike.com/ (Accessed: 26.04.2021).

[10] *ESM+Strategy*, Available at: https://www.esmgrp.com/strategy (Accessed: 26.04.2021).

[11] *Spring Bean vs. EJB – A Feature Comparison*, Available at: https://www.baeldung.com/spring-bean-vs-ejb Accessed: (28.04.2021)

[12] *Spring Framework - Overview*, Available at: https://www.tutorialspoint.com/spring/spring_overview.htm Accessed: (28.04.2021)

[13] *2020 Java Technology Report*, Available at: https://www.jrebel.com/blog/2020-java-technology-report (Accessed: 28.04.2021)

[14] *A Comparison Between Spring and Spring Boot*, Available at: https://www.baeldung.com/spring-vs-spring-boot (Accessed: 26.04.2021)

[15] *Testing Microservices — Java & Spring Boot*, Available at: https://www.pinterest.at/pin/48202658494862103/ (Accessed: 27.04.2021)

[16] *Java Programming - The State of Developer Ecosystem in 2020 Infographic — JetBrains: Developer Tools for Professionals and Teams*, Available at: https://www.jetbrains.com/lp/devecosystem-2020/java/ (Accessed: 26.04.2021)

[17] *Client-side vs. server-side rendering: why it's not all black and white*, Available at: https://www.freecodecamp.org/news/what-exactly-is-client-side-rendering-and-hows-it-different-from-server-side-r (Accessed: 28.04.2021)

[18] *TIOBE Index for April 2021*, Available at: https://www.tiobe.com/tiobe-index/ (Accessed: 26.04.2021).

[19] *The JAVA language concept*, Available at: http://aladin.u-strasbg.fr/java/JavaInfo.htx (Accessed: 26.04.2021).

[20] (April 2, 2021) *Oracle Java SE Support Roadmap*, Available at: https://www.oracle.com/java/technologies/java-se-support-roadmap.html#products (Accessed: 26.04.2021).

[21] *SQL vs. NoSQL: Comparative Advantages and Disadvantages*, Available at: https://bitnine.net/blog-computing/sql-vs-nosql-comparative-advantages-and-disadvantages/ (Accessed: 27.04.2021)

[22] *MongoDB and MySQL Compared*, Available at: https://www.mongodb.com/compare/mongodb-mysql (Accessed: 27.04.2021).

[23] *5 Popular NoSQL Databases Every Data Science Professional Should Know About*, Available at: https://www.analyticsvidhya.com/blog/2020/09/different-nosql-databases-every-data-scientist-must-know/ (Accessed: 27.04.2021)

[24] *PostgreSQL vs MySQL: The Critical Differences*, Available at: https://www.xplenty.com/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/ (Accessed: 27.04.2021)

[25] *How To Develop and Build Angular App With NodeJS*, Available at: https://medium.com/bb-tutorials-and-thoughts/how-to-develop-and-build-angular-app-with-nodejs-e24c40444421 (Accessed: 28.04.2021)

[26] *AngularJS*, Available at: https://en.wikipedia.org/wiki/AngularJS (Accessed: 28.04.2021)

[27] *Github Topics - react*, Available at: https://github.com/topics/react (Accessed: 28.04.2021)

[28] *Top 10 In-Demand Web Development Frameworks in 2021*, Available at: https://towardsdatascience.com/top-10-in-demand-web-development-frameworks-in-2021-8a5b668be0d6 (Accessed: 28.04.2021)

[29] *Tutorial: Using Thymeleaf*, Available at: https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html (Accessed: 27.04.2021)

[30] *Who is using Thymeleaf?*, Available at: https://www.thymeleaf.org/whoisusingthymeleaf.html (Accessed: 27.04.2021)

[31] *Color wheel - color theory and calculator — Canva Colors*, Available at: https://www.canva.com/colors/color-wheel/ (Accessed: 13.05.2021)

[32] *The fundamentals of understanding color theory*, Available at: https://en.99designs.pt/blog/tips/the-7-step-guide-to-understanding-color-theory/ (Accessed: 13.05.2021)

[33] *Color Picker by WebFX – Free HTML Color Chooser*, Available at: https://www.webfx.com/web-design/color-picker/ (Accessed: 13.05.2021)

[34] *Binding a List in Thymeleaf*, Available at: `https://www.baeldung.com/thymeleaf-list` (Accessed: 22.05.2021)

[35] *Mission, Vision & Values — Erasmus Student Network*, Available at: `https://esn.org/mission-vision-values` (Accessed:13.05.2021)

[36] *Home - Socrative*, Available at: `https://www.socrative.com/` (Accessed: 11.05.2021)